

CompTIA SY0-401 Security+  
TABLE OF CONTENTS

Section 0: Introduction

0.1– Introduction -----8-11

Section 1: Network Security

Routers, Firewalls, and Switches -----12-14

Load Balancers and Proxies -----14-16

Web Security Gateways and UTMs -----16-17

VPN Concentrators -----17

Network Intrusion Detection and Prevention -----18-19

Protocol Analyzers -----19-20

Spam Filters -----20-21

Web Application Firewalls -----22

Application-Aware Security Devices -----23

1.2 – Network Administration Principles

Firewall Rules -----24-26

VLAN Management ----- 26-27

Secure Router Configuration -----27-28

Access Control Lists -----28-29

Port Security and 802.1X -----29-30

Flood Guards -----31-32

Spanning Tree Protocol and Loop Protection -----32-33

Network Separation -----34

Log Analysis -----35

1.3 – Network Design Elements and Components

DMZ-----36

Subnetting the Network -----37

VLANs-----38

Network Address Translation -----39-40

Remote Access-----40-41

Telephony -----41

Network Access Control-----42

Virtualization-----42-43

Cloud Computing-----43-45

Defense in Depth-----	45-46
<b>1.4 – Common Protocols and Services</b>	
IPv4 and IPv6-----	46-47
IPsec -----	48
ICMP and SNMP-----	48-49
Telnet and SSH -----	50-51
Transferring Files-----	51-52
DNS-----	52
HTTPS and TLS/SSL -----	53
Storage Area Networking -----	54-55
NetBIOS -----	55-56
Common Network Ports -----	56-60
Protocols and the OSI Model-----	60-62
<b>1.5 – Troubleshooting Wireless Security</b>	
Wireless Encryption -----	62-63
EAP, PEAP, and LEAP -----	63-64
MAC Address Filtering -----	64
SSID Management-----	65
TKIP and CCMP -----	65-66
Wireless Power and Antenna Placement-----	67
Captive Portals -----	68
Antenna Types -----	68-69
Site Surveys-----	70
VPN Over Open Wireless Networks-----	71
<b>Section 2 – Compliance and Operational Security</b>	
<b>2.1 – Risk Related Concepts</b>	
Control Types-----	72-73
False Positives and False Negatives-----	73-74
Reducing Risk with Security Policies-----	74-75
Calculating Risk-----	76-77
Quantitative and Qualitative Risk Assessment-----	77-78
Vulnerabilities, Threat Vectors, and Probability-----	79-80
Risk Avoidance-----	80-81
Risks with Cloud Computing and Virtualization-----	81-82
Recovery Time Objectives-----	83-84
<b>2.2 – Integrating Systems and Data with Third Parties</b>	
On-boarding and Off-boarding Business Partners-----	84-85
Security Implications of Social Media-----	85-86
Interoperability Agreements-----	86-87
Privacy Considerations with Third-Parties-----	87-88

Risk Awareness with Third-Parties-----88-89

Data Ownership and Unauthorized Data Sharing -----89

Data Backups with Third-Parties-----90

Security Policy Considerations with Third-Parties----- 91-92

Third-Party Security Compliance -----92

**2.3 – Risk Mitigation Strategies**

Change Management-----93

Incident Management-----94

User Rights and Permissions-----95

Security Audits----- 96

Data Loss and Theft Policies----- 97

Data Loss Prevention----- 97-99

**2.4 – Basic Forensic Procedures**

Order of Volatility-----99-100

Capturing System Images----- 100-101

Capturing Network Traffic and Logs----- 101-102

Capturing Video----- 102-103

Recording Time Offsets----- 103-104

Taking Hashes----- 104-105

Taking Screenshots----- 105-106

Interviewing Witnesses----- 106

Tracking Man-Hours and Expenses----- 107

Chain of Custody----- 107

Big Data Analysis----- 108

**2.5 – Incident Response Procedures**

Preparing for an Incident----- 109-110

Incident Identification----- 110-111

Incident Escalation and Notification----- 111-112

Incident Mitigation and Isolation----- 112-113

Lessons Learned from Incidents----- 113-114

Incident Reporting----- 114-115

Incident Recovery and Reconstitution ----- 115-116

First Responder----- 116

Data Breaches----- 117

Incident Damage and Loss Control----- 117-118

**2.6 – Security-Related Awareness and Training**

Security Policy Training and Procedures----- 118-119

Personally Identifiable Information----- 119-120

Information Classification----- 120

Data Labeling, Handling, and Disposal----- 121-122

Compliance Best Practices and Standards----- 122-123

User Habits-----123

New Threats and Security Trends----- 124

Social Networking and Peer-to-Peer Security----- 125

Gathering Training Metrics----- 125-126

**2.7 – Physical Security and Environmental Controls**

HVAC, Temperature, and Humidity Controls----- 126-127

Fire Suppression----- 127-128

EMI Shielding----- 128-129

Hot and Cold Aisles----- 129

Environmental Monitoring----- 130

Physical Security----- 130-133

Physical Security Control Types----- 133-134

**2.8 – Risk Management Best Practices**

Business Impact Analysis----- 135

Critical Systems and Components----- 136

Redundancy and Single Points of Failure----- 137-138

Continuity of Operations----- 138

Disaster Recovery Planning and Testing----- 139-140

IT Contingency Planning----- 140-141

Succession Planning----- 141

Tabletop Exercises ----- 142

Redundancy, Fault Tolerance, and High Availability----- 143-145

Cold Site, Hot Site, and Warm Site----- 146

**2.9 – Security Goals**

Confidentiality, Integrity, Availability, and Safety----- 146-148

**Section 3 – Threats and Vulnerabilit**

**3.1 – Malware Types**

Malware Overview -----149-151

Viruses and Worms----- 151-154

Adware and Spyware----- 154-156

Trojans and Backdoors ----- 156-159

Rootkits----- 159-160

Logic Bombs-----160-161

Botnets----- 161-162

Ransomware----- 163

Polymorphic Malware-----163-164

Armored Virus-----165

3.2 – Attack Types

Man-in-the-Middle Attacks -----165-167

Denial of Service Attacks----- 160-169

Replay Attacks----- 170

Spoofing----- 171-172

Spam----- 172-173

Phishing----- 174-175

Vishing----- 176-177

Christmas Tree Attack----- 177-178

Privilege Escalation----- 179

Insider Threats----- 180-181

Transitive and Client-Side Attacks----- 181-182

Password Attacks----- 182-185

URL Hijacking----- 186-187

Watering Hole Attack----- 187-188

3.3 – Social Engineering Attacks

Shoulder Surfing-----188-189

Dumpster Diving----- 189-190

Tailgating----- 191-192

Impersonation----- 192-193

Hoaxes----- 194-195

Whaling----- 195-196

The Effectiveness of Social Engineering----- 197-198

3.4 – Wireless Attack Types

Rogue Access Points and Evil Twins----- 198-199

Wireless Interference----- 200-201

Wardriving and Warchalking----- 201-202

Bluejacking and Bluesnarfing----- 203-204

Wireless IV Attacks ----- 205-206

Wireless Packet Analysis----- 207-208

Near Field Communication----- 209

Wireless Replay and WEP Attacks----- 210

WPA Attacks----- 211-212

WPS Attacks----- 212-213

3.5 – Application Attack Types

Cross-site Scripting-----2013-216

SQL Injection, XML Injection, and LDAP Injection-----216-217

Directory Traversal and Command Injection----- 218-219

Buffer Overflows and Integer Overflows----- 219-220

Zero-day Attacks----- 221-222

Cookies, Header Manipulation, and Session Hijacking-----222-225

Locally Shared Objects and Flash Cookies-----225

Malicious Add-ons and Attachments----- 226-227

Arbitrary and Remote Code Execution----- 228

**3.6 – Mitigation and Deterrent Techniques**

Monitoring System Logs-----229-230

Operating System Hardening----- 231-233

Physical Port Security----- 234-235

Security Posture----- 235-236

Reporting----- 237-239

Detection vs. Prevention----- 239-240

**3.7 – Security Threats and Vulnerabilities**

Vulnerability Scanning Overview----- 241-242

Assessment Tools----- 243-245

Assessment Types----- 245-246

Assessment Techniques----- 247-248

**3.8 – Penetration Testing and Vulnerability Scanning**

Penetration Testing-----249-251

Vulnerability Scanning----- 252-254

**Section 4 – Application, Data, and Host Security**

**4.1 – Application Security Controls and Techniques**

Fuzzing----- 254-255

Secure Coding Concepts----- 256-257

Application Configuration Baselining and Hardening----- 258-259

Application Patch Management----- 260-261

SQL and NoSQL Databases----- 262-263

Server-side vs. Client-side Validation----- 263-264

**4.2 – Mobile Security Concepts and Technologies**

Mobile Device Security----- 264-267

Mobile Application Security-----267-269

Mobile BYOD Concerns -----269-271

**4.3 – Establishing Host Security**

OS Security and Settings----- 271-272

Anti-Malware -----272-275

Patch Management----- 276-277

White-Listing vs. Black-Listing Applications -----277-278

Trusted OS-----279

Host-based Security----- 280-281

Hardware Security----- 281-282

Host Software Baselining-----283

Virtualization Security----- 284-286

**4.4 – Ensuring Data Security**

Cloud and SAN Storage Data Security-----287-288

Data Encryption----- 288-290

Hardware-based Encryption----- 291-292

States of Data----- 293-294

Permissions and ACLs-----294-295

Data Policies----- 295-296

**4.5 – Static Environment Security**

Embedded System Security----- 297-298

Static OS Environments----- 299-301

Mitigating Risk in Static Environments----- 301-303

**Section 5 – Access Control and Identity Management**

**5.1 – Authentication Services**

RADIUS and TACACS -----303-304

Kerberos-----304-307

LDAP and Secure LDAP----- 307-309

SAML-----310

**5.2 – Authentication, Authorization, and Access Control**

Identification, Authentication, and Authorization----- 311

Authorization and Access Control----- 312-313

Single Factor Authentication----- 313-314

Multi-Factor Authentication ----- 313-316

One-time Password Algorithms----- 316-317

CHAP and PAP-----317-319

Single Sign-on----- 319-320

Federation and Transitive Trust -----321

**5.3 – Account Security Best Practices**

Roles and Account Credentials-----322-323

Group Policy-----323-324

Managing Password Policies----- 324-325

Privileges----- 326-327

User Access Reviews and Monitoring----- 327-328

**Section 6 – Cryptography**

**6.1 – General Cryptography Concepts**

Cryptography Overview-----329-331

Symmetric vs. Asymmetric Encryption----- 331-332

Public Keys and Private Keys-----333-334

Session Keys----- 334-335

Block vs. Stream Ciphers----- 335-336

Transport Encryption----- 337-338

Non-Repudiation----- 339-340

Hashing----- 341

Key Escrow----- 342

Steganography----- 343-344

Elliptic Curve and Quantum Cryptography----- 345

Perfect Forward Secrecy----- 346

**6.2 – Cryptographic Methods**

WEP vs. WPA----- 347-348

Cryptographic Hash Functions----- 348-350

Symmetric Encryption Ciphers----- 350-351

Asymmetric Cryptography Algorithms----- 352

One-Time Pads----- 353-354

NTLM----- 354-355

Transport Encryption Algorithms----- 355-358

Strong vs. Weak Encryption----- 359-360

**6.3 – PKI and Certificate Management**

Certificate Authorities----- 360

Key Revocation ----- 361

Digital Certificates ----- 362

Public Key Infrastructure----- 363-364

Key Recovery----- 364-365

Public and Private Keys----- 366-367

Key Registration ----- 367-368

Key Escrow----- 369

Trust Models ----- 370-371

## **Introduction to CompTIA SY0-401 Security+**

CompTIA's Security+ is a popular IT certification, and the certification topics range from networking to cryptography. In this video, I'll give you an overview of the Security+ certification and I'll give you some tips to help with your exam testing experience.

Hello, everyone. Welcome to Professor Messer's CompTIA SY0-401 Security+ Plus training course. I'm James Messer, and I'll be your host through all of these videos that we've made available for you, absolutely free, to learn everything you're going to need to know to pass your Security+ exam.

First, let me tell you a little bit about myself. I've been working in the technology industry for over 25 years now, and I've had jobs that have ranged from delivering printer cables to working in mainframe and supercomputer environments and, of course, managing day to day operations of networks and security. This is one of the things I like about the Security+, is that once you have this certification, you can really apply it towards so many different aspects of information technology. And of course, I am also CompTIA certified as well. So as you go through this course, you'll know that you're learning this information from someone who has also gone through and passed this exam.

When you start going through my video course, you may find that the structure is a little bit different than other video courses you may have seen before. I like to keep my videos relatively short. I focus on a single topic or set of topics.

So you'll find that most of the videos are 15 minutes or less. I try to make them very quick so that you can get in, get the information you need. And then you can either stop and go do something else, or you can go directly to the next video.

It was important to me when I started creating this series that the videos themselves would be worthwhile. I not only wanted to create absolutely free videos that anybody could watch, but I wanted those videos to be very high quality. It was important, as someone who's done corporate training in the past and someone that has had technical roles that needed to get certifications, that I was providing you with something that would be worthwhile to watch.

We've designed these videos for everyone, whether you're looking for your first job. Maybe you're looking to move up in your current career, or maybe you're changing industries and now getting into technology. We think that this video series should work perfectly for you.

And as I mentioned, this is an absolutely free course. Not just one video is free or two videos are free. Every single video of this course can be watched online for absolutely free.

There's no registrations involved. There's nothing extra that you need to do. You simply go to the web page. You click on the video. And you can watch every minute of every video.

If you do want to take these videos offline, I do have versions that you can purchase so that if you wanted to be on a plane or away from your internet connection, you would still have access to all of this training material. I've also taken all of the content of my videos and created a relatively inexpensive study guide that you can purchase. This has all of the text and all of the graphics.

All of the displays that you see are in this downloadable PDF file. That way, you can concentrate on watching the videos instead of taking notes, and all of your notes, when you're ready to study for that final exam, will be in one place for you. You can find information on the offline videos, the downloadable PDF study guide, and every single video in the entire index can be found at my website at [ProfessorMesser.com](http://ProfessorMesser.com).

CompTIA stands for the Computing Technology Industry Association. It is our industry's largest provider of these vendor-neutral IT certifications. It used to be in our industry that we would have to go to individual manufacturers to be able to take their specific exam. What CompTIA has done is create a vendor-neutral version of that so that one single exam can take the place of all of those manufacturers' exams.

You'll also notice that CompTIA is a worldwide organization with reach to over a hundred different countries. You can not only take the Security+ exam in English, but also in many different languages around the world. CompTIA certifications are some of the most popular and well known certifications in our industry. If you're someone who is looking for a job, one of the things you'll notice is that CompTIA certifications comes up again and again when you're looking at job descriptions.

Many people also use CompTIA certifications to improve their own career. You can get your Security+ and move into, perhaps, a security group within your own existing organization. And some organizations won't even hire you unless you have certain CompTIA certifications. A very good example is the federal government of the United States that requires certain certifications to even hold a security job within their ranks.

And of course, your goal may not be any of these things. It may just be that you're getting your certification for your own personal satisfaction. This latest version of the CompTIA Security+ exam is the SY0-401. If you're looking at your study materials and the books that you use, make sure that they all are referring to this particular version number of the exam.

If you're working in an educational facility that is a CompTIA Academy partner, you might also see this exam referred to as the JK0-022. It's exactly the same exam with exactly the same questions and the same content. But the numbering scheme is a little bit different if you are a CompTIA Academy partner.

The Security+ exam is 90 minutes long, and you'll get a maximum of 90 questions. The bulk of these questions will probably be traditional multiple choice questions. But CompTIA will also ask you performance-based questions. These questions may be a matching set of questions. You may have to rank things in a certain order. But they are outside the scope of what you would traditionally see in a multiple choice exam.

The passing score on the Security+ exam is 750 on a scale of 100 to 900. That's a little bit different than a percentage complete. And it's difficult to calculate that, because we believe that CompTIA rates their different questions as having a different value. So you're never quite certain the number of questions that you would have to get correct to be able to pass the exam. You just have to do the best you can and hope that you got some of those high valued questions correct.

The exam includes topics from six different areas or domains. The first domain is network security. And 20% of your exam will be associated with topics from that particular domain.

Domain 2 is compliance and operational security, for 18% of the exam. Threats and vulnerabilities is another 20% of the exam. And then you have application, data, and host security at 15% of the exam. And section 5 is access control and identity management, for 15%. And finally, cryptography is 12% of the exam.

You'll get a randomized set of questions from all of these domains to make up the total of 100%. And you'll want to check with CompTIA's exam objectives to know exactly what's

expected from each one of these domains. That's why I always give my first tip to anybody who's taking a CompTIA— is to go to the CompTIA website and download those exam objectives. They're very detailed, and they will let you know exactly what you need to learn to be able to pass this certification exam.

Of course, I've also got these videos available for you to watch. And I've also got these videos integrated into a book from GTS Learning. So you can watch a little bit of a video and read all about it on the same page of these ebooks. There are also topics on my website, in my forums and on my online chat, where you can discuss all of these topics with other people who are also taking these exams.

There are a lot of different testing centers for the CompTIA exams. Some are better than others. So you may want to familiarize yourself with the testing centers in your area and perhaps visit some of them before you book your exam.

You want to get a lot of sleep and be ready to take that exam so when you walk in the door, you're able to focus on everything that's expected of you. And if you can arrive on site a little bit early, you may also be able to relax, be able to look over your notes, and in some cases, even start the exam before your allotted time.

Sometimes there's a lot of anxiety when you're taking an exam like this. So when you sit down in the testing center in front of your computer, before clicking that Start button, you may want to take a few deep breaths, maybe write down some notes on the information that the testing center is giving you, and see if you can't become more comfortable with the environment that's around you. If during your exam there is some noise in the outside or the environment is not comfortable, you can stand up and let the testing center know that there are problems. It's much better for the testing center to resolve these issues during your exam than to have you go all the way through the exam and not do as well because there were problems with the environment.

The CompTIA exam allows you to move throughout any question on the exam at any time. So you can jump all the way to the end. You can look at questions in the middle of the exam. And then you can go all the way back to the very first question if you'd like to.

If you run across a question that you're having a problem with, you can mark it and move to another question. And then before you submit the exam, you can look at a list of all of the questions that you've marked, and jump back to them at any time to address any questions that you might have had. So it's a good way, from a time management perspective, to skip over the ones that you're really having a problem with, make sure you're able to answer the ones you really know, and then easily find those to jump back to later.

Before you submit the exam to be graded, you'll get a page that lists out everything on the exam— all of the questions. It will show you what your answer was for those questions. And it will tell you if you have any questions that have been marked. So before you're able to submit that, make sure you look over that, and be sure that you've answered every single question that's on the exam.

Once you hit that Submit button, there's no going back. Your exam will be graded, and you'll know immediately whether you passed or whether you failed. If you pass, congratulations. But if you've failed, don't walk out without a sheet of paper that tells you how you did on the exam. It will take each of the Security+ domains and tell you how you did on each one. If you're planning a retest, it's useful to have that information so you'll know what sections you need to really study for next time.

So you can start studying right now for your Security+ exam. Go out to my website at [ProfessorMesser.com](http://ProfessorMesser.com), and start watching those videos. If you need a companion book, you can find that at [ProfessorMesser.com/gtslearning](http://ProfessorMesser.com/gtslearning). I've also got all of the videos

arranged by all of the CompTIA domains in exactly the same order as the exam objectives. So it's very easy to find the topics you're looking for.

I've also got online message boards that you can use to communicate with others who are taking the exam. And of course, these are also absolutely free to participate in. I've also got a real-time chat that's at the bottom of every page of my website, so you can interactively communicate with all of the other students that are on the Professor Messer website. And I like to hang out in there as well and answer questions whenever I can.

Well, this should be your start to learning everything you need to know for your Security+ exam. I hope you're excited to get started. And we wish you the best of luck with your studies.

**Tags:** [certification](#), [comptia](#), [free](#), [overview](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## Routers, Firewalls, and Switches – CompTIA Security+ SY0-401: 1.1

The **fundamental technologies** in almost every network are **switches**, **routers**, and **firewalls**. In this video, you'll learn how these devices are used to connect and protect our network devices.

Let's start our discussion of these network devices on switches. These switches are really great big bridges. They operate **at Layer 2 of the OSI Model**. I put a representation, here, of the different layers of the OSI Model, here, on the left-hand side.

We're really talking about Layer 2, primarily, in these switches. All of these switches do all of this switching, this **MAC Layer Look-Up**, in hardware, so they're really, really fast. And what's nice about the back planes of these devices is they can communicate to each other. Two devices on this device communicate to each other directly, without having to use any bandwidth or bother anybody else that's on the network.

So they're very, very efficient in the way they operate. They decide where traffic goes, based on the data link control address of a device. And most of the time on an Ethernet network, for instance, that's the **MAC address**.

The network card address of the different devices. So there's a big table inside of these machines that understands exactly everybody who's plugged into this device. And whenever it needs to decide which way packets go, it looks to see what the destination **MAC address**.

It references back the big table of lists, and it says oh, that particular device is on port seven. I'm going to send the traffic over to port seven. There are many, many, many ports on these enterprise devices.

They really are the core of an enterprise network. If you're in a large, or even a small environment, and you're plugged into the network, you're probably plugged into a switch in almost every situation. This happens to be a really, really large switch with lots of slots, and you can fill it up with many different kinds of ports.

Some switches are very small. They're workgroup switches, and there may be many of those stacked up inside of a closet, for instance. But most of the time, when you're on somebody's network, you're on a switch.

You'll also see, if you ever look at a network diagram, a switch represented with this diagram, here, where you've got arrows just pointing left to right or up and down. They don't go any other direction. They pass straight through a particular device that represents that **Layer 2** switching, where we're just sending traffic on its way.

You also are able to have a lot of bandwidth go through these devices, and this becomes a little bit of a challenge from a security perspective. You have so many different devices plugged in. You have so much data going back and forth.

How do you begin to manage traffic, especially understand the security relationship between two devices that may be talking to each other on the same switch? And that is a bit of a challenge. We have to now layer our security, not only inside devices like this, but also on the end stations themselves and the servers. If we ever want to be able to see everything end-to-end, that's really the only way to go about doing it.

Since switches operate at **Layer 2**, everybody's on the same subnet. So to be able to separate our network into other pieces, we need something to be able to move up to a higher level, the **OSI Layer 3**, and that would be a router. And usually routers are in the center of the network.

And most of the time, they're connecting all of these different switches to each other. Perhaps connecting an internet connection, as well. Any time you have to connect two different **IP subnet**, you're going to need a routing function somewhere. This may be on a standalone device, or it may be part of a **software module** or **hardware** module within a switch.

So you'll sometimes hear the term a **Layer 3 switch**. That's really talking about a router that is embedded, or installed, inside of a switch. You're not really switching at Layer 3. You're really routing at Layer 3.

You'll also see this represented on network diagrams as these different arrows that are pointing in different directions. So if you ever see that **90-degree angle** on an arrow going through a diagram, it's probably referring to a router. If you ever hear the term **Layer 2**, you can think switching.

If you hear the term Layer 3, you can think routing. And that's usually how we're representing it. Sometimes we don't say we need to route, sometimes we say we need to do Layer 3 between those two particular subnets.

These are also able to connect different network types. So you'll connect a **Wide Area Network connection**, a **fiber-based network connection**, a **copper based network connection**, and they'll all go through the router. And the router's smart enough to do whatever types of signaling translations, or any type of packet translations, between those different networks.

So not only are we connecting different IP subnets together, we can connect very, very diverse networks together with routers. It provides us a lot of functionality to be able to do that in our enterprise environments. Usually also from a security perspective, there is a little bit of filtering capability in here.

You have the ability to filter out certain port numbers. A very, very basic filtering functionality. In the security world, we tend to do only a very basic type of filtering in our router, because we'll use a **firewall** to be able to do a much more efficient job of protecting our networks. If you are ever working around network people, they tend to want to have the **switches switch**, the **routers route**, and have the **firewalls do firewalling**.

If you try to combine some of these things together, not only it is complicated, but a router doesn't really make a good firewall. So that's one of the nice things about keeping these as separate components is that you can manage them much better from a security perspective. Firewalls really cover the security perspective for the rest of the stack of the **OSI layer**.

We've talked about switching at Layer 2, we've talked about routing at Layer 3. Well, at Layer 4 and all the way up to Layer 7, we have firewalls. And firewalls are really our first and last line of defense when that traffic is going in and out of our network.

If we need to protect servers, we need to protect our users, we need to separate ourselves from the big bad internet, it's a firewall that's going to be doing that. This can be also a device that is able to encrypt data into and out of the network. Very often, we'll connect firewalls to each other, and we'll build encrypted tunnels between those connections.

We'll talk a lot about encryption technologies, and the way that we do these tunnels in other parts of these videos that we look at. But it's usually the firewalls that are the endpoints between the two. You may have a firewall at your home office. You may put a firewall at a remote site. You may connect them together through the internet.

And in order to keep your data private, as it goes through that public internet, we can create encrypted tunnels between the two, and essentially, send all of our data between those two sites, all encrypted. Even if somebody was to look at that data going by, they

wouldn't be able to make any sense of it as it's going through. Many firewalls can also act as proxies.

**Proxies** is a very, very traditional method of separating internal networks from the internet. **Proxies** work by making a request to a web server, but instead of talking directly to the web server, you're really talking to the proxy that you have inside of your network. That proxy then takes your request and makes the request on your behalf.

When it receives the response from that web server out there the internet, it looks through the content, and makes sure there's nothing bad in there. Usually makes sure that that's something you're allowed to look at, and then it sends you the response. By putting that right in the middle, it is separating the internal network from the internet.

There's some nice security benefits to doing that. Most firewalls that you're going to find it can also be Layer 3 devices. So you will very often see the firewall on the edge of the network as the internet is coming into it.

And it's performing routing for us, and it's doing network address translation for us. So many times you don't have to have a Layer 3 router right behind it. The firewall's simply doing all of that routing for us.

And because it's right there on the edge, it can route to the internet, it can route to a **DMZ**, it can route to our internal network as if it was a standalone router all by itself. Think of it as having routing functionality with all of these great firewalling and security technologies built right into the technology.

**Tags:** certification, comptia, firewall, router, security, switch

**Category:** CompTIA Security+ SY0-401

### **Load Balancers and Proxies – CompTIA Security+ SY0-401: 1.1**

If you need to expand the capacity of your applications and network resources, you'll need to use technologies like **load balancers** and **proxies**. In this video, you'll learn how load balancers and proxies can be used to increase the scale of your network capacities.

As web technologies became more and more popular, we found that we needed some way to scale these web servers that we were using. When you go to google.com, you don't go to a single server. Google obviously has hundreds and hundreds, and perhaps even thousands of Google servers out there that we are connecting to at any particular time.

So out there on the network, you're usually hitting something like a load balancer. There's many ways to distribute load across different servers. **A load balancer** is a very, very common way to do that when you're in a data center. It's usually a piece of hardware that is in the rack, and it's connecting to four different servers in my particular picture, but it can be many, many more servers on the network.

The **load balancer** receives the request from your browser, and it distributes the loads evenly, usually, across these servers. You can decide how exactly you'd like to distribute that load, and it really is distributing across what we call a cluster of different servers. The idea is that I don't really care which server I'm connecting to. All four of those servers are exactly the same.

When I hit a web page. I just want to be able to have the accessibility to the web page, and by distributing that load across them, we can be assured that we've got some uptime and availability that we are happy with. We don't want things slowing down so much that we don't want to use that web server. Obviously you're going to need this in a really large environment, because usually you've got thousands and thousands of people connecting to your website all at the same time.

The **load balancers** become very, very important in those environments. And you can distribute based on the load, you can distribute based on what content. Maybe one of these servers provides images, another one provides video, another one provides the web page itself. You can decide exactly how to separate the load across those.

This creates a little bit of a security challenge for us, because you have all of these people coming into the load balancer. You want to be sure that it's being distributed across all those servers, are all of the servers updated with the latest security patches? Are there vulnerabilities that have not been addressed on the different servers. They are different machines, so it becomes very important that you keep all of them updated to the latest security patches.

And of course, you want to be sure there's no security issues by using the load balancer itself. Somebody was to find an exploit that would manipulate how the load balancer worked, it could essentially send the data somewhere else other than your web server, and you certainly don't want that to happen, either. Another very common security technology that we use to protect their end users from bad things on the internet is a **proxy**.

A **proxy** is a server or series of servers that's designed to sit right in the middle of your users and the big, bad internet, and its job is to take any requests the user's sending out to a web server and stop it, and then send the request on its behalf. So what will happen is you'll be on your machine, you'll need to go to Google, and you'll send your request.

And instead of it getting out to Google, your proxy server sitting in the middle, it says wait, hold on, before you can go to Google, I'm going to stop you right there, and I'm going to find out what you need and ask Google myself. And a **proxy server** makes the request to Google and receives the response. The **proxy server** then looks at the data and makes sure there's nothing bad inside of there.

There's no malware, there's no viruses. Usually makes sure that the user's even allowed to use Google itself, and if it likes the results, it will then send the answer back down to the end user. So it's an extra step between you and the internet, and there's some performance requirements there. Obviously sitting in the middle and stopping everybody's internet connection requires that that proxy server be pretty beefy.

Able to handle a lot of different connections and a lot of bandwidth going across the internet. Proxy servers are also very useful for caching. If I'm going to a website and I'm downloading a big file, and the next person on the internet does exactly the same thing, proxy servers are often configured to cache information.

And so if they see a second request come through, they can simply send that information directly to the second user, and they don't have to make that request back to the internet. And therefore the results are getting a lot faster to the end users, and there's a lot of bandwidth we also did not have to use to go out to the internet. So some nice performance increases if the proxy server is doing caching.

There's two ways to really configure the way that your systems use the proxy server if a proxy server is an **explicit proxy**, and that is one where you must configure your browser and your other applications to know that the proxy's there, and to use the proxy.

Then you'll need to make sure you make those changes in your browser, or you need to make sure as a security administrator that you're finding an automated way to make those changes inside people's browsers, and of course any other application that needs to access the internet. There's also another type of proxy you can choose to use called a **transparent proxy**.

That means you don't have to configure anything for your end users. You don't have to change any of the settings in your browser, you don't have to change settings on your

third party apps, but sometimes applications will not work properly through a **transparent proxy**. The proxy is still proxying, and so there are changes being made to the network communication, so not all applications work very well with **explicit proxies** or **transparent proxies**.

And from a security perspective, it becomes a bit of a challenge for us. If not all applications can use a proxy, and yet the **proxy** is the primary way that we provide additional security to the internet, then maybe we're opening ourselves up, because we end up having to make exceptions for certain applications. And any time you make an exception in a firewall, any time you make an exception in a proxy, you're opening a little bit of a window there for bad things to occur.

So that's the balancing act you have to make as a security professional. Do you use a proxy to provide a little bit more security, or do you provide a different methodology to allow filtering to and from the internet? There's a number of options available. Proxies is simply one of many that you can choose from.

**Tags:** certification, comptia, load balancer, proxy, security

**Category:** CompTIA Security+ SY0-401

### **Web Security Gateways and UTM's – CompTIA Security+ SY0-401: 1.1**

As technology has improved, we've added more and more functionality to our security gateways. In this video, you'll learn about unified threat management appliances and the functionality they bring to securing our network resources.

So far we've talked about routers and switches and firewalls and **WAFs**. We're going to talk about other devices in the next module. One of the things that you'll start to see in our industry, though, is the ability to collapse a lot of these functions into a single device. This is especially useful if you have a remote location, and you don't have a lot of room, or you don't have a lot of budget to be able to buy many, many different devices for those sites. You'll hear this referred to as a **Unified Threat Management device**, a **UTM device**. You may also hear it referred to as a **Web security gateway** because very often it is the single gateway between a remote office and their access to the internet.

Inside of these devices you may have things like **URL filters** or content inspection engines that's determining what website you're going to and determining how that website is categorized. Is it an auction site? Is it a search engine site? Or is it perhaps a category of site you should not be visiting. It can allow or prevent access based on that. Many of these devices will also look for things like malware and spyware and viruses. They'll go through your emails that are going back and forth and determine if it's spam or not. There may also be functionality in there at the networking level to directly connect you to your provider through a **CSU/DSU**.

You could also see routers and switches are very common to have in these particular appliances. And of course— it's an all-in-one appliance— it has to have a firewall inside of it as well. Occasionally you'll also have things for additional security like **IDS** or **intrusion prevention systems** as well.

There's a lot you can put in a single device. Now whether it does all of these things well or not is a different question, because you can also add on to that some network functionality to be able to shape traffic. Maybe you want to limit the bandwidth that's being used for people doing streaming media and still allow traffic to go through for your Voice Over IP or your critical internet connectivity or critical applications.

That's a lot to go into a single device. And very often these devices do suffer a bit for performance, and they suffer a little bit by functionality. Being a master of many things is a difficult prospect for any device. But usually you can get away in a small office with

having a subset of these things, and at least being your first line of defense against some of these things coming into these locations.

So as you're looking at what you're buying or what you may be using in your environment, look to see, is a **UTM**? Is it doing many different functions? And you'll be able to determine what of that **UTM** you'd be able to use for security in your environment.

**Tags:** bandwidth

shaper, certification, comptia, csu, dsu, firewall, ids, ips, malware, qos, router, security, spam, switch, utm, web security gateway

**Category:** CompTIA Security+ SY0-401

### **VPN Concentrators – CompTIA Security+ SY0-401: 1.1**

We're an increasingly mobile workforce, so we therefore need technologies to keep us secure while we travel. In this video, you'll learn how **VPN concentrators** can be used to provide encrypted tunnels from our favorite coffee shop to our corporate network.

**VPN concentrators** are becoming increasingly common. You can buy them now for your home office, even, and use **VPN software** to connect back through an encrypted tunnel to your home office, where you can then print on your local printer, even though you're somewhere else.

The way these concentrators work is out there on the internet, you may be at a coffee shop, you may be at a hotspot somewhere, and you want to be able to communicate to your corporate or your home network, but you don't want the people on the internet to see what's going on. In steps the **VPN concentrator**.

This is exactly what we'll do, and by using some software in your operating system or on your machine, you're able to create an encrypted tunnel through the internet to the **VPN concentrator**. And it may not just be one person, it may be many, many different people, in some cases hundreds or thousands of people that are connecting through these encrypted tunnels back to **the VPN concentrator**, and it's creating now that virtual private network.

Because all of that's encrypted, even if somebody did get their hands on these packets going back and forth, they wouldn't be able to do anything with them, because all of the data inside of those packets is protected. The **VPN concentrator**, then, is doing a lot of hard work. It is decrypting this traffic. It's putting it onto the internal network, on this green network, and you're able to communicate as if you were sitting in the same building as all of these devices.

And as the response goes back to your machine, the **VPN concentrator** is in charge of **encrypting that data** and sending it across that link again. The process of **encrypting** and **decrypting data** is very, very CPU intensive, so very often these **VPN concentrators** are very, very hardware specific devices, so they can keep up with the speeds that we need to be able to use, because many times you have hundreds or thousands of people coming in.

You've got a lot of encrypting and a lot of decrypting to do as that traffic goes by. If you are in a remote location or you or someone who is very mobile, and needs to communicate back to your home office, a **VPN concentrator** is a practical necessity in today's security environments.

**Tags:** certification, comptia, concentrator, security, vpn

**Category:** CompTIA Security+ SY0-401

## **Network Intrusion Detection and Prevention – CompTIA Security+ SY0-401: 1.1**

**IDS** and **IPS technology** can watch for a wide variety of attacks by examining the traffic as it passes through the network in real-time. In this video, you'll learn about **IDS/IPS technology** and the identification technologies that they use.

It's nice to be able to look at every packet that goes through the network and be able to see the details of exactly what's going by. But there's a lot of traffic that's going through our enterprise networks and there's no possible way that a human being would be able to analyze all of that traffic and be able to find the bad stuff within it. That's why we created technologies called the **Intrusion Detection Systems or Network Based Intrusion Prevention Systems**. These systems are designed to do that for us, to watch the traffic go by.

And if it sees something in there, it will detect that a vulnerability or some type of bad traffic is inside those traffic flows. In the case of an **IPS**, those **IPS's** are designed to actually stop that traffic. So it's looking through all of this traffic going by to identify things that are known exploits against our operating systems, things like buffer overflows, or cross-site scripting. These types of very, very well known vulnerabilities that people try to use to gain access to your system using other means that they should not be.

When we have these **IPS** and **IDS systems** on the network, we usually call it an **IDS** or an **IPS** because of what it can do when it finds a problem. **IDS** an intrusion detection system is designed to alarm or alert should it see something bad on the network. But generally, IDS's can't stop anything.

And obviously if something bad's going across your network, you may want the option to be able to stop that traffic. And that's where IPS's step in. An **intrusion prevention system** is designed that whenever it sees something bad on the network, it stops it right there— never gets inside your network, never makes it to the end user, and therefore, makes your network a little bit better from a security perspective.

One of the challenges you have obviously then with **IDS's** and **IPS's** is identifying things properly. If you have an **IPS** in place, you're providing that prevention and you're dropping those packets. You need to be very, very sure that you're not dropping legitimate traffic. And that's a balancing act we as security professionals have all the time. We want to stop the bad stuff, we want to allow the good stuff, and we have to find a happy medium in between the makes everybody happy.

A fundamental technology used in **IPS's** and **IDS's** is something called a **signature-based based match**. We want to be able to look at the exact code going over the network. And if we see this code, then we're going to stop that traffic. Some of these signatures can be very, very detailed.

This is one for a worm, a **conficker-a**. This is the shell code and I pulled this right from an open source **IPS type system** called **snort**. And you can see one of the signatures that used to gather that information. Pretty complex, very, very detailed. In this case, all in hexadecimal. We're looking for some very specific kinds of data going through.

But once we have these particular signatures in place, we're just looking for an exact match. If we see anything exactly matching that data going through the network, we'll stop it right there. Another type of detection is one called an **anomaly based detection**. This is one where we would have a probe or device on the network looking at what is normal. And it builds a baseline of what it thinks is normal on the network.

If the network or certain aspects of metrics go well above that normal range, it becomes an anomaly. Something that normally you wouldn't see on the network and it might provide an alarm to you to let you know that this particular thing just happened. We had suddenly a lot more people than normal try to log into the network. We suddenly had huge bandwidth spikes out on our network connection that goes well beyond what we normally

might see on our network. So it gives you the security professional a little bit more information. A few more metrics that you can use to start to understand is something wrong going on out there and should I do something about it.

Another method of watching that's very, very specific is something called a **behavior-based technology**, one where we're trying to watch anything that someone might be doing. Did someone log into a server? When they logged into the server, did they run a certain command? After running that command, did they try to perform another certain type of command?

And if we see these things happen one after the other after the other and identify a certain behavior of what someone is doing, maybe then we'd like to get an alert to find out why somebody's logged into a machine and performing those particular commands. Probably one of the most common beyond the signature-based forms of detection is one called **heuristics**. And this is really an emerging technology that we've had for a number of years and we continue to make it better and better and better.

**Heuristics** is a bit of artificial intelligence. We're seeing a type of traffic flow come into the network and we'll watch it for a little bit and see if it changes. And we'll try to look for different aspects of the way traffic flows might operate. So we're applying some intelligence to these traffic flows and then based on what we're seeing, provide you with more information about whether this particular packet flow is one that contains good types of data or perhaps bad types of data.

Obviously, **heuristics** is one that is constantly evolving. It's one that we're getting better and better with. And occasionally, heuristics can provide us with the wrong types of information too. It really depends on how good that heuristics engine is to be able to check for that.

So obviously for heuristics, you can have widely different qualities of heuristics out there with different products—very, very different from a signature. From a signature match, you know yes or no that absolutely matches the signature. Whereas, heuristics there's a lot more gray in between, it's not really a black and white type of situation.

**Tags:** alarm, alert, anomaly, behavior, certification, comptia, heuristics, intrusion detection, intrusion prevention, security, signature

**Category:** CompTIA Security+ SY0-401

### **Protocol Analyzers – CompTIA Security+ SY0-401: 1.1**

If you ever want to know exactly what's happening on a network, then you need a **protocol analyzer**. In this video, you'll learn about protocol analyzers and how they can be used to monitor traffic and solve network problems.

If you recall from the CompTIA Security Plus requirements we looked at at the beginning of this video, one of the things it asked us to know about was a sniffer. And it has a little bit of a double meaning in our industry, it's sort of a generic term we're now using. But the term sniffer is actually a product, it's actually a registered trademark of a company called **NetScout systems**.

It's a product that's been around for a long time, and because it has such longevity in our industry, we've almost used the name generically as a device that is able to capture packets from the network and provide us with analysis and decodes of that information. So if you're using the term sniffer, you're really referencing a product line from another company.

It's almost becoming the xerox and the Kleenex of the network security industry. There are very common ways to capture packets and display those packets on the screen, and

that's really what we're talking about. When you hear somebody say sniffer, what they're really talking about is a **network analyzer**. It's something that can grab those packets and show them in plain English on the screen what's going through the network.

Take all those ones and zeroes, and those signals that are going across that ethernet connection, and somehow put them all together and show us that that was a web conversation, or exactly what might be going across that link. Very, very common technology. In fact, these days it's almost an easy one to find, because there are some very good open source options.

One of the most popular network analysis tools you'll find is one called Wireshark. If you go to [wireshark.org](https://www.wireshark.org), you'll be able to download that, load it on your machine, capture your packets from the network right now, and be able to see exactly what's going on on your protocol analyzer.

**Tags:** [certification](#), [comptia](#), [packets](#), [protocol analyzer](#), [security](#), [sniffer](#), [traffic](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Spam Filters – CompTIA Security+ SY0-401: 1.1**

Few things are more frustrating than having to sort through an email inbox that's full of unsolicited email. In this video, you'll learn about spam filters and some common methods used to separate the legitimate email from the spam.

No one likes to receive unsolicited emails. It's one of those things that can fill up your inbox, and receiving all of this spam also creates overhead on your email servers and of course the bandwidth coming into your organization. You probably have something like this in your environment, where there is information coming in from the internet, and all of those emails would go to a central mail relay on the inside of your network that then sends that information off to the mailboxes that you ultimately access.

So this is a great place to be able to stop the spam. If we stop it at the mail gateway or the mail relay, then we can prevent it from ever showing up in your inbox. And there's a number of different ways to do it. Some people will do it on the mail relay itself, other folks prefer to outsource that to the cloud. There are many companies that will provide for email filtering before it's ever sent to your mail relay which means that you can spend time on your system managing other parts of your email, rather than dealing with spam filtering.

There are many methods that these spam filters used to determine whether an email message is legitimate or whether it is simply unsolicited spam email. One very common way is to have a white list. You would only be able to receive emails from people that were on the list. If you're not on the list, then those emails are never delivered into your inbox.

Another way to analyze a piece of email to determine if it's spam or not, is to examine the protocol itself that's used to transfer the mail from one mail gateway to another. We do this using **SMTP**. This is the **simple mail transfer protocol**, and there are certain standards that are used to transfer this. If it's a spammer, they may not be using the exact

standards. And when you look at the details of what's being sent, you may be able to filter out email because it's not directly following those standards.

Another determination you can make is to look at the sender of the email, and then compare that to the IP address of who's actually sending it. If you're expecting an email that's coming from an associate who's in the same country as you, and yet the IP addresses from somewhere halfway across the world, a reverse **DNS** can easily start showing those discrepancies. And perhaps it might be a decision to categorize that as spam, rather than something that might be legitimate.

The process of sending a mail between mail servers is completely non-interactive. There are no human beings at either end of that particular line of communication. So if there are delays that occur, we never really see those as the end user. One of the things the spammers are doing, of course, is sending as many emails as possible in a very short period of time.

So one way to frustrate that process is to do something called **tarpitting**. **Tarpitting** is an intentional slow down of the communications process between these mail servers. So as the spammer's trying to send that mail in, you just take another couple seconds before you ever reply back to what they're doing. The **spammers** are expecting that machine to respond back in milliseconds, and instead you're taking many thousands of milliseconds to respond. And in some cases, the spammer will stop the process and move on to someone that they can use a very fast transfer method against.

So by simply slowing down the conversation, you may be able to prevent a lot of the spam you might normally get. In some cases, the spammers are simply making up names and sending them to email addresses in your domain. So instead of filling up your mail server with a lot of unknown users, or perhaps responding back to the spammer saying you don't have the right name and they can simply try something else, you can just simply block it right there.

Filter out anybody who is not a legitimate recipient and instead of informing the sender that name doesn't exist, they don't hear anything back from you, and the spammer has no idea whether they're spam email ever made it to the end user or not.

Whether you're filtering out your spam with an on-site server or you're using a cloud based service, these methods can help you eliminate a lot of the spam that might be incoming to your environment.

**Tags:** certification, comptia, filter, reverse dns, security, smtp, spam, tarpitting, whitelist

**Category:** CompTIA Security+ SY0-401

## **Web Application Firewalls – CompTIA Security+ SY0-401: 1.1**

Firewalls now examine port numbers and applications as they traverse the network, but what's protecting our servers from malicious user input? In this video, you'll learn how a **web application firewall (WAF)** can protect from attacks that take advantage of unexpected application use.

A newer type of security technology that we've seen over the last few years is something called a **web application firewall**. You'll hear this referred to as a **WAF**. A **web application firewall** is looking at web conversations, and it's trying to determine based on that web conversation if the information within your packets, within that conversation, is legitimate.

You'll often see this used to make sure that when people are inputting information into a web form that that information is correct. If you're trying to put in a serial number, or the date, or a **ZIP code**, this particular web application firewall technology is looking to see, is that really is zip code you're adding in there? Is that really a serial number?

The reason that's important is that if you try to put unexpected information into one of these fields and you're able to manipulate the application, you can often find exploits that might give you direct access to the database that's contained behind it, or direct access to the web server on which this particular application is running.

So by having this additional check of that input data, you're hopefully protecting against things like database injections and things like buffer overflows. And those are very bad things, because often that does allow somebody some very detailed access to some very sensitive data. You want to try to avoid that.

Because of these web application firewalls' ability to look and validate this input, it can prevent things like sequel injections. The very crafty hackers will go into a field that is supposed to be for a **ZIP code** and they'll, instead, add special characters and their own sequel commands to try to gain access to the raw data in the database. Now the only way they be able to do that is if the application wasn't written well, and it's allowing some of these types of input.

But even if the application isn't written well, having this web application firewall gives you another line of defense. You may not be able to check all the different ways to validate data inside of your application, but your web application firewall certainly can. And it can check and make sure that somebody's a trying to do a **SQL** injection right there at your ZIP code field and it will prevent that data from blowing through and on to your database.

You see this a lot in things called the **payment card industry data security standard**, the **PCI DSS**. If you do a Google search for that, you'll see a lot of information about that because you don't want people have access to credit cards, and so the payment card industry came up with a series of standards that people have to follow if you store credit card information on your servers.

One aspect of the **PCI DSS** standard is that you have to have web application firewalls, because if somebody is going to that ZIP code field and they are typing in something that would give them access to the sequel database, then they would also have access to, potentially, credit card numbers that might be in that database. So you can start to see why having something at the application level, to be able to validate input into those web fields, becomes critically important, especially when sensitive data is involved.

**Tags:** certification, comptia, security, waf, web application firewall

**Category:** CompTIA Security+ SY0-401

## **Application-Aware Security Devices – CompTIA Security+ SY0-401: 1.1**

Today's modern security devices not only must understand the network, but they must also be fluent in the language of our applications. In this video, you'll learn about security devices that can protect your network by watching application flows.

Today's modern security devices are looking at everything that goes by on the network, and they are examining it based on the applications that you might be using. This is talking about the **OSI models application layer**, and really examining every bit of data that goes through the network. No longer are we just interested in port numbers or protocols that might be going by. We're really interested in the entire application.

These are called many different names, it might be an **application layer gateway**, a **stateful multilayer inspection device**, it might be something using deep packet inspection. But all of those really mean the same thing. We're looking at every single bit and every single byte that's going over the network. We're examining it, we're doing a protocol decode, we're looking at what application that's associated with. And in some cases we're looking at the way the application is working as it communicates back and forth between devices,

As you can imagine, this is a very advanced piece of hardware and software because it's looking at more data than we ever looked at on the network. Every packet has to be analyzed. It has to be categorized.

There is also a number of security decisions that have to be made. Is this a legitimate application? Is the application transferring a file? Does that file happen to have any malware inside of it? Is that application even allowed on this network? Those security decisions can only be made if you really are looking at every bit and every byte that goes by on the network.

The latest generation firewalls on our network are application aware. They can look at all the traffic going by and categorize those flows based on what the application happens to be. So the firewall knows if there is **Microsoft SQL Server traffic**. It knows if there's Twitter traffic, if there's YouTube, or there's BitTorrent. It allows the security person to make the decisions on what applications are allowed and what applications should not pass through that firewall.

**Intrusion prevention systems** have also taken hold of this application layer view of the network because they can create much more detailed and accurate signatures. **IPSs**, of course, are trying to stop someone from the outside trying to take advantage of a vulnerability inside of a server or an application, so it's useful for those particular devices to also be aware of what applications are being used to provide that level of access onto your network.

And of course the firewalls that are inside all of our hosts are **Windows firewalls**, for instance— are very aware of the applications their running on our computers. So it knows if you're using a browser, it knows if you're performing an **FTP** it knows if you're using homegroup, or file and printer sharing, and you're able to make security decisions based on what application is there, and if that application should allow access from somebody else that's on the inside or outside of your network.

**Tags:** [application-aware](#), [certification](#), [compbia](#), [firewall](#), [intrusion prevention](#), [ips](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Firewall Rules – CompTIA Security+ SY0-401: 1.2**

How does a firewall know which traffic is good, and which traffic is bad? In this video, you'll learn about the fundamentals of firewall rules and you'll step through an actual firewall rule set.

When you work in technology, and especially in security with technology, there's a lot of rules involved. These rules are procedural rules, they're rules about the way that our devices operate, they're rules about allowing people access to different things. A lot of the things that we do, really, are procedural. They're both technical and non-technical in nature.

You need policies and procedures, and what happens when somebody walks into your building and they don't have their badge? You need policies and procedures when a new person is hired. What is the process for getting them the proper credentials to the network?

And of course, there's technical procedures as well. When somebody is asking for those credentials, how do you provide those credentials? There's got to be a way that you provide that into your system.

They're for environments that are dealing with finances or medical or other sensitive types of data. There are also a lot of very specific rules as well. Some of these rules are things that you've created internally, some of these rules have been created by third parties and are required of you to maintain because of the types of environments that you have. If you're someone who has medical information– that's very private information– there are certain rules and requirements that you must follow to be able to protect your patients' data.

When we look at our technical devices like firewalls, there's packet filters, even email systems have filtering rules within them. These are, in some cases, very very technical rules. We're dealing with bits and bytes, and so we also have to apply those rules to what we're doing as well. Sometimes, in fact almost all the time, our technical rules are going to follow the procedural rules that we set up.

When somebody is asking for credentials to be able to log into the network they might fill out a form, they might visit a web page and add their information, and then you've now got to provide the technical back end to provide them that data. So you're almost always spending a lot of time– before you even touch a keyboard– determining what the process is going to be. Determining what somebody has to provide you in order for you to get them the access that they need.

These are things that, hopefully, you're figuring out based on your requirements for your business or your organization. But very often you're taking into account a lot of different requirements from a lot of different people and putting that all together to create the technical answer that you might need.

Let's start a technical discussion of rules dealing with firewall rules, because as a security professional you are going to be using firewalls quite a bit, and you're going to be going through the rule bases to allow or deny people access to certain resources. And as we look at the rules of the firewall, we're usually making the decisions on what people can do based on a number of different tuples.

These are different categorizations of data that we'll then add information to. So we may decide if somebody is coming from a certain source **IP address**, they may be going to a

certain destination IP address, maybe using a certain kind of port number. It may be a certain time of the day, they may be using a certain application, and on and on and on.

Different firewalls have different tuples that you could use to make this determination. And we'll group that together and say if you match all of these put together, then you are either allowed or denied access to certain resources. Usually, there is also a logical path that you follow with firewall rules. Almost always, you start at the top of a rule base and you work your way down.

It's not that way with every firewall, however. You'll need to look at your particular firewall and how it operates to find out exactly the path that it follows, to be able to determine whether somebody has access to the internet or not. These rules can also be very generalized. Maybe you might want to set a rule that says if you're anybody inside the network, you can surf the internet. That's a very general rule.

Or it might be very specific, that says if you are in the marketing department and you are coming from a particular source IP address, then you have access to this particular resource on the internet that is that another IP address. That's a very specific rule. So you tend to put those specific rules at the top of your firewall list so that they're fired on first if it applies, and then other more general rules are at the bottom.

In almost all firewalls— this is not always the case— but a good firewall, anyway, I like to think that there is something called an **implicit deny** at the very bottom of that list. And that means that if it goes through your list of rules and at the very bottom of the list it hasn't hit any of those rules, we're just going to drop the traffic. It is implicitly denied traffic at the bottom.

Some people will put an explicit deny at the bottom. They'll create a rule at the bottom of their firewall that says if it's any-to-any type traffic at the bottom, deny everything. Sometimes that's useful just so you can see it, and know that that rule is being fired on. Sometimes they're doing it so that it gets logged, because usually implicit denies don't log traffic.

Can you imagine logging everything that comes in from an internet connection that's not intended to come inside of your network? It would be an enormous amount of traffic. Some people, however, would like to see that information. So they may put an explicit deny down at the bottom of the rule base, just so they can capture and log some of that information coming by. If you don't put a rule, then it's probably the case that your firewall has an implicit deny, and it's going to drop all that traffic anyway.

Let's step through a very simple firewall rule base, and let's see what's really involved here. I grabbed this rule set directly from an internet service provider. This is their default configuration for their web servers on their Linux host. And you can see they're numbered one through seven, all in order. In fact, they start from the top and work their way down.

And you can see there is a default policy here. This particular rule set has an implicit deny, which means unless you're allowing it in this list, it gets denied.

So let's start with rule number one, which says if you're coming from any remote IP address on any remote port number, and you're connecting to this particular web server on port 22 with the **TCP protocol**, we're going to allow that. If you're a really good person about documenting your particular firewall rules— there may be one of these fields that's a description field— and you may put in here that this rule allows anybody to be able to **SSH** to our particular firewall.

This is how you take those well-known port numbers and apply them back to certain applications that are used. Now that that matches, we'll then allow an **SSH**. So let's go down to the next one, let's say the traffic coming through doesn't match that, then we'll

examine this rule. It says from any remote IP to any remote port number over port 80 that's running **TCP**, allow that traffic.

And of course, port 80 **TCP** is **HTTP** based traffic, usually. That's our web service traffic. So if we're running a web server on this machine, somebody's trying to connect to it with a browser, it is the firewall rule number two that will allow that traffic to connect to this server.

Let's do one more. Rule number three is **remote IP** is any talking to any remote port number over port 443. That is the **TCP protocol**. Allow it, and of course, port 443 is **HTTPS**. And you would step through this list and make sure that everything here is what you would like it to be.

In fact, the next rule that says allow all IPs from any port number to **local port 8443** over the **TCP protocol**, allow it. That's not one you often see. **8443** is not usually a well known protocol. That is a protocol that is used— a port number that's used— to open up access to the management part of the web server. So if you don't want people managing your web server with that front-end web based management that you've created, you may want to deny traffic if it's coming from any remote IP address.

So that's a good example of how you can allow or disallow the traffic based on any of these port numbers coming through. And we're simply following the rules of our firewall, one after the other, until it either fires or gets to the bottom where traffic is implicitly denied.

**Tags:** certification, comptia, firewall, implicit deny, rule, security, tuple

**Category:** CompTIA Security+ SY0-401

### **VLAN Management – CompTIA Security+ SY0-401: 1.2**

**VLANs** are an essential part of nearly every enterprise network. In this video, you'll how VLANs work and how VLANs are used to segment and organize our networks.

If you've done a lot of networking, then you've certainly done a lot of **VLAN segmentation**. These **virtual LANs** give you a way to separate out your IP subnets into logically separate areas. Even though it's all running on a single switch, none of those devices on the different **VLANs** can communicate to each other unless there's a router involved. From a networking perspective, we're usually doing this because we want to separate out things from an IP addressing perspective.

From a security manager's or a security administrator's perspective, we're often doing this so that we can separate out different parts of the organization. You might want to put the **HR department** on one **VLAN**. You might want to put the shipping and receiving department on a completely different **VLAN**. So now you're allowing your firewall, or your router, or your firewall that is acting as a router, to be the gatekeeper, to prevent the HR people from directly communicating to the folks in shipping and receiving and vice versa. There's maybe sensitive information on the **HR servers**, and that would give us yet another way to provide some control over the traffic going back and forth over our network.

We're usually grouping people together in these VLANs by function. It doesn't have to be that way, but that's usually how it turns out— the finance department, the executive team, the HR department, et cetera. You don't want to have people, though, separated too far away from the resources they need to use. If there's a central email server, you want to have centralized access for that because everybody's going to be communicating to that email server. You don't want to put the email server on the HR department's VLAN, and then force everybody else to come into that VLAN to have access to that particular resource.

These **VLAN** communications and automatically putting people into different **VLANs** is very often integrated with our **Network Access Control**. You recall the last video we did, we talked about getting access to the network using **802.1X**. Once you get your credentials and you're authenticated on to the network, your Network Access Control system can be set to automatically put you in the correct VLAN. It doesn't matter what switchport you happen to be a plugging into. That's pretty flexible, and from a security perspective really provides us with the way to make sure that we're keeping people segmented onto the VLANs that are very specific to their job function.

Without any way to manage where somebody's plugged in on a VLAN, it becomes a little more difficult to manage. If all we had was a single switch and everybody on that switch was on one VLAN, we would not only have to logically separate, but really physically separate everybody on to their own switch. You would have red VLAN on one switch, the green VLAN on another switch, and the blue VLAN on a third switch. And if we wanted to communicate between those, everybody would go up to a router and then be sent down to the VLAN they wanted to talk to.

With VLAN management, we can mix and match. We can have different VLANs on different switches. We can assign, either automatically or manually, individual ports to be members of certain VLANs. We can have a port for the green VLAN here. On another completely different switch, a port for the green VLAN here. And because we're building these communication links between the two switches— called trunks— we're able to send green VLAN information between all of those devices, and blue VLAN information between all of those devices without having to route anything.

If the green VLAN did want to talk to the blue, they would need to route between those to get back, so there are some network requirements to think about how you would deploy this. But from a security perspective, it gives you a lot of flexibility on where you put people logically in the environment and still protect those very critical resources from other departments.

**Tags:** [certification](#), [comptia](#), [nac](#), [security](#), [vlan](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Secure Router Configuration – CompTIA Security+ SY0-401: 1.2**

A **router** is one of the most important devices in your network. In this video, you'll learn how to securely connect and manage your routers and other infrastructure devices.

When you're working with routers, you're working with firewalls, you're working with other devices. One of the challenges you have is that you're providing very sensitive security information to those devices.

You don't want anybody to have access to that, except for you and the other security professionals in your environment. Every device works a little bit differently on how it stores data, and how it secures data.

So you'll want to look at your router, or your firewall, or your switch, to determine— how is that information stored, and how can I communicate to that device in a way that's not going to provide a lot of data in the clear that somebody might be able to see.

Very often, in fact, you'll find that devices on your network still have the default username and password on them. You'll see this a lot for people's home routers. They don't think of changing them. Username— admin, login— admin. Username— administrator, login— administrator. It's a very, very common thing, unfortunately, to see this— even in some of the largest networks in the world.

So you really have to go through and audit, make sure that you have changed all those passwords. It's so quick to pass by that when you're installing a device for the first time.

You also have to think about how you're communicating to the device. Are you communicating over a channel that's in the clear traffic? You certainly don't want that. You want to communicate to this device in an encrypted form. So you're going to want to use things like, **SCP**, which is an encrypted method of doing a file copy. Or **HTTPS**, which uses encrypted communication to these devices. Or **SSH**, if you need to be able to communicate to the device as well.

**TFTP**, probably not so much. That is "in the clear" traffic. If somebody wanted to see the configuration files which you're transferring via **TFTP**, all of your very sensitive security information would be in there— all of the information you added for IP address ranges, and the different resources available.

If you got your hands on a firewall, or a router log, or a router configuration, there's a lot of good information in there, that bad guys would be able to use.

You also have to think about the way that your configurations are being stored. Are they stored on the device, are they stored in an encrypted form?

Are you transferring them, and putting them in a network location as a backup? Is the backup encrypted? You don't want people getting access to that. Is the backup secure? Is it on your local workstation? Do you put it on a flash drive, is it on a network drive?

You need to think about what you're doing with this data— both on the device, and once you take it off the device. And if you just keep those things in mind, you can be certain that then your network infrastructure devices— and the configuration on those devices— is going to remain secure.

**Tags:** [certification](#), [comptia](#), [https](#), [router](#), [scp](#), [security](#), [terminal](#), [tftp](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Access Control Lists – CompTIA Security+ SY0-401: 1.2**

**Access control lists** are used in almost every security device on your network. In this video, you'll learn how **ACLs** are used to provide secure access to your important resources.

Another way to allow or disallow access to resources, is through something called an **access control list**. You may hear this referred to as **ACLs**, or as **ack-ulls**. The **access control** lists are things that you would assign to an object, or a network, that would allow permission to that object, or that network.

You see access control lists used in many different environments— in file systems, network devices, operating systems. It's a very, very common way of setting up permissions to particular resources.

For instance, this is the way you do an access control list, to be listing out permissions of a file, for instance. In this access control list, Bob can read the files, Fred can access the network, James can access the network, 192.168.0.1/24, using TCP ports 80, 443, and 8088. So you can have very generalized access controls in your access control list, or very specific access controls in your access control list.

Here's a good example of an access control list that might be inside of a firewall. Here's access list. All of this is a member of the first access list in the firewall.

Here's a deny access control that says, if you are on 172.16.5.2 with this mask— which means, really, this IP address— we're going to deny any traffic from that IP address. So that locks it out. Here's another one, access control list we're going to deny 172.16.5.3. So the first two of these rules in this access control list are denying traffic to individual IP addresses.

And here's the last access control— that says, permit any. Which means, if you aren't 172.16.5.2, 172.16.5.3, everybody else is allowed to communicate.

That's a very simple access control list, but it gives you an idea of— looking at a router, or looking at a firewall— the process it goes through to go down that list, and allow or disallow access to resources on the network.

**Tags:** access control list, acl, certification, comptia, security

**Category:** CompTIA Security+ SY0-401

## **Port Security and 802.1X – CompTIA Security+ SY0-401: 1.2**

If you need to secure a physical network port, then you'll want to consider some type of **network access control (NAC)**. In this video, you'll get an overview of port security and I'll show you a step-by-step of 802.1X in operation.

Another challenge we have is security of our switches and the ports that are on our switches. When you have all of these different ports on a device, anybody can walk into a conference room, they can walk into an empty jack that might be wired up on your network, they can plug in their device, and they might have access to all of the internal resources of your organization. Because of that, there's a type of security called **Network Access Control, NAC**. You'll sometimes see this referred to as **Port-based Network Access Control**. And what it's really referring to is a standard technology called **IEEE 802.1X**.

The idea is that before you're really allowed access to that switch port and it's turned on and giving you access to the network, you first have to authenticate. And that way someone couldn't walk into your conference room, plug in, and see your network because they wouldn't have the authentication credentials. It uses some technologies called **EAP**, "**eep**", and **RADIUS**. This stands for an **Extensible Authentication Protocol**. It's a very standard way to authenticate on a network. And **RADIUS** is a way to store and communicate authentication details like user names and passwords that stands for **Remote Authentication Dial In User Service**. We'll almost refer to this all the time as **EAP** or "**eep**" and a **RADIUS** server somewhere in your environment.

And we're talking about protecting here the physical interfaces of your switch. When somebody talks about port-based network access control, you sometimes have to make sure— Are you talking about the physical ports on my switch? Or are you talking about **TCP** and **UDP ports**? In this particular example for switch port security and 802.1X, we're really talking about physical access to the ports on these switches.

You can also— and this is a very good best practice— is to administratively enable and disable ports as they are needed. If a port is not being used, disable it. That way you won't accidentally enable it. And that also means somebody can't walk up to your switch and plug in and get access to your network.

You also have switches that have some intelligence built into the ports themselves. If somebody was to get on your network and duplicate **MAC addresses** on your network in an effort to redirect traffic, your switch can recognize when some of those things occur and stop people from what we call **spoofing**, or trying to fool the switch into thinking that you are somebody else on your network. And by having these switches look for those types of security issues, you can prevent also someone from stepping in and taking over a session that might already be authenticated. Your switch recognizes— wait a second, that **MAC address** already exists somewhere else. I'm not going to allow you access onto this switch.

Network access control using **IEEE 802.1X** is a pretty complex set of protocols, and needs to be. We're providing access to ports on a switch through software. And so there needs to be a lot of checks and balances in between. There's three major components you have to think about with 802.1X. There's something called a **supplicant** that is a piece of software that is running on your computer that recognizes how to communicate via 802.1X.

There's also something your supplicant will be talking to called an **authenticator**. And as you might think, that's the device that really is providing a middle ground for authenticating on the network.

And then there's something called an **authentication server**. Your supplicant never really talks directly to the authentication server. Your supplicant on your computer is talking to the authenticator, who's then passing through that information to the authentication server.

The conversation goes something like this. When you first connect to a network, you don't have access to the network. You're not able to communicate. You're essentially in a initialization phase where you have no ability to communicate out on the network. And you sit there, and you wait. Very often, the authenticator, the authentication device that's on the switch— usually it's a part of the switch itself— is sending out these requests every so often saying, hey, is there anybody new out there? This is really called an **EAP**, an “**eep**” request, to find out— has somebody else plugged in lately? And if you have, you might want to let me know that you're here.

In fact that's the next step, is that the supplicant recognizes, oh, somebody's asking for me. Yes, my name's James. I'm here on the network. I'd like to go ahead and start the process of gaining access to the network. The authenticator then lets the authentication server know that, by the way, James is on the network. What would you like to do with this? And the authentication server is now waiting to find out more information from that.

So the authenticator asks, hey, James, are you able to talk? Can you communicate on the network? The authentication server would like to communicate. Well, sure, here's my credentials. Let me give you information about who I am and how I can communicate on this network. And the authentication server then receives that information from the authenticator that says, hey, James is on the network. Here's his credentials. Maybe this is someone you might want to allow access to the network.

The authentication server looks through that information, says that looks fantastic. James is allowed access to the network. Go ahead, enable some ports, and let him on to the network. And at that point your machine is allowed access to the ports on your network. Very often the switch is reconfigured to allow access to a particular **VLAN** or to the **VLAN** that is specific to you, and now your computer is on the network and able to communicate.

It's a relatively complex process, but as I mentioned, it's one that really provides us with a lot of checks and balances and ensures that only the people who are allowed on the network really have access to the network.

**Tags:** [802.1x](#), [certification](#), [comptia](#), [nac](#), [port security](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Flood Guards – CompTIA Security+ SY0-401: 1.2**

A relatively easy way to overwhelm network devices is to attack them with a flood of network packets. In this video, you'll learn about some of the more popular methods of flooding a network and how to help prevent these denial of service attacks.

One very common network administration function for security professionals is looking for and trying to stop floods on the network. You often see these identified on intrusion prevention or intrusion detection systems, and that's because those devices are always on the network. So they're able to keep track of just how much traffic is coming into the network, and just how much there is of what type.

You're really interested in floods because of something called **DoS** or **DDoS**, and that stands for **Denial of Service** and **Distributed Denial of Service**. That's a bad term. You don't want services that you're providing on your website, the applications you're providing to third parties, to be denied to them.

And unfortunately floods that are coming through the network from many different locations can very quickly overwhelm the technologies that we might have for a web server or a firewall or our routers, and because of that we have to keep track of this.

And if we happen to see a large number of **Denial of Service** or **Distributed Denial of Services** that are coming from many different places, then we'll need to somehow start affecting that in some way. Turning them off, making sure that the services that we're providing continue to be available to everyone.

One very common type of flood is a **SYN flood**. If you recall from your network plus studies, a **SYN** is the first packet sent to a server. There's that three way handshake that occurs for **TCP**. There's a **SYN** that is sent, you get a **SYN-ACK back**, and then you would send an acknowledgement that is the third packet.

Well, if a system is simply sending SYNs, there's a possibility that they could now overwhelm the sessions that are available on a server. A web server, a file server, any type of server, really.

So keeping track of just how many SYNs are coming in can give you an idea on if somebody's really trying to attack you or not. And if the number of SYNs suddenly spikes up and starts using up more resources than you would like inside of your servers, or inside of your firewall, or inside of your router, you now have something that you can do to prevent some of that from happening.

So you need to look at those opportunities that people might have to flood your network with SYNs and keep track of that as just another thing that would be available. Normally your firewalls are going to block that right at the firewall, so hopefully your services will not be affected.

Another common type of flood that you would see on the network is a **ping flood**, or a **ping scan**. If somebody's outside of your network and they're just trying to find out are there some devices turned on on your network? Because as soon as I find the device, I'm going to try to find an exploit that will get me into the device.

So a common first step is for someone to send pings around to try to find those devices. If they really want to be malicious, they might turn up the volume of those pings, if you will, where they start flooding your network with pings and requiring that your devices now send back responses to those. And of course, that just creates more and more traffic on your network and quickly overwhelms those devices.

So if you start to see ping floods on your network, again, identified by your **IPS** or your **IDS**, that's another thing to look for. Maybe somebody's doing reconnaissance, or if it's a lot of them, maybe it's somebody trying to bring down a part of your network.

Another type of flood that you'll see is a **port flood** or a **port scan**. Once somebody identifies a machine, they might want to know what services are running inside of that machine. Is it a web server, is it an email server, is it a time server? What is it?

I want to know more about it, because if I can figure out what service is running on that server, I can then cross reference the versions of what might be running there to find out if there's any known exploits.

If it's an email server, I can send some carefully crafted email messages to attempt to bring it down.

So there's things that are going to happen at every step along the way. The **SYN floods**, the **ping floods**, the **port floods**, that would either provide somebody with information about what's happening on the network or provide a way for people to stop the services, deny the services, that are on those particular devices.

In any case, you want to be able to identify them as quickly as possible, and then find ways to mitigate this coming through, whether that's from a firewall or from the server itself, and turn off some of those floods that are hitting your network.

**Tags:** broadcast, certification, comptia, flood, ping flood, ping scan, port flood, port scan, security, storm, syn flood

**Category:** CompTIA Security+ SY0-401

### **Spanning Tree Protocol and Loop Protection – CompTIA Security+ SY0-401: 1.2**

**Spanning Tree Protocol (STP)** is an important standard that provides a mechanism for switched networks to avoid outages due to network loops. In this video, you'll learn how loops can cause network issues and I'll demonstrate what happens when Spanning Tree constantly adjusts to avoid network loops.

A very common way to create problems on a network is to build a loop, is to connect two switches to each other and then connect them to each other again, and watch the packets start circling between them as fast as they can go. And as they go by, more traffic gets on to the network, and more traffic starts looping. And eventually you completely overwhelm your infrastructure devices just because of all the packets that are looping back and forth. And the only way to resolve it is to break the loop, wherever it happens to be.

Hopefully that's not something that is happening normally. And you have to be very careful when you start plugging in devices to your switches to make sure that a loop is not going to occur, because if one happens then you have big problems. And you're going to know very, very, very quickly that a loop has occurred, because your entire network is going to come to a screeching halt.

Fortunately, we built mechanisms and protocols within things like our switches and our bridges to prevent these things from happening. These Mac layer protocols themselves have no way to know if they're in the middle of a loop, so what we've done is put the intelligence on the switch or on the bridge. And we use a standard called **IEEE 802.1D**. This is something called spanning tree that prevents loops.

And one of the nice parts about spanning tree is that it is very much a standard that everyone uses. Maybe it's not called spanning tree— maybe a manufacturer has taken spanning tree and has done a little bit extra to it to make it a little faster, or to change the way it operates a bit. But it's all really based on this spanning tree technology that was created by Radia Perlman, and it's really used everywhere. Every switch, every bridge you're going to run into has some methodology to prevent loops, and it's really built on the fundamentals of **IEEE 802.1D**.

One key aspect of the **Spanning Tree Protocol** is that all of your bridges on your network, or your switches, can all talk to each other. And most of the time that's exactly the way your network is set up. In the Layer 2 mode all the devices can see everybody else.

There's **three types of ports in a spanning tree technology**. There is a **root port**, and that's the port that talks back to the root bridge. One bridge on the network is the **root bridge**, and it's usually the one with the smallest Mac address number associated with it, or one that you would designate as the root bridge. Here's Bridge 1 at the top of my list. It is designated as the root bridge. It does not have a root port because it is the root— it doesn't need a link to the root. What it does have are designated ports, which are ports that are available to send traffic out over the network.

And as you can see this network is very much interconnected. It would be very, very easy to have a loop appear if spanning tree wasn't in place. But every bridge knows of everyone else's. If it knows where the root bridge is, it creates an open port to the root bridge with the root port, it creates a link to the rest of the network that's your designated port, and then it recognizes that there's a potential for a loop, and creates a **blocked port** so that traffic will not go out of that connection. Now if you're in this type of scenario and you need to get, for instance, from Bridge 21 if you're on Network C and you need to get to Network B, you're going to have to go all the way back out to the bridge and back down again to get to Network B because these particular ports on Bridge 21 and Bridge 11 are blocked.

Well, this is great. Everybody knows about each other. They keep track of each other. Messages are sent very often between these bridges. But what if something happens? What if there's a problem? For instance, you have a break right here in the network. If you wanted to get to Network Y from Network C, normally you would go all the way around, and it would take this connection all the way through the network. But if you can't get to Bridge 6, now you have no way to get down to Network Y.

Now since all these bridges are talking to each other, they would immediately go into a mode where they decide— wait a second, I can no longer communicate down to Bridge 5. I need to find some way in Bridge 6 to get that direction. And so what happens is it reconfigures itself on the fly and changes. The root port now on Bridge 5 swaps over so that now it passes through Network Y, and Bridge 11 recognizes that and gets rid of its blocked port there. And now Network C and Network Y are directly connected to each other, and we still maintain communication from Network A back up to the root or anywhere else on the network if we want to.

This is a critical piece to how this spanning tree technology works. There's a lot of details underneath the surface. This is a very high level view. But you can see here this is a great way to prevent loops. And it's also a great way to create redundancy in your network, and if you do happen to have an outage, still maintain the availability of what's happening. From a security perspective this also maintains uptime and prevents those loops for bringing down your network and creating a denial of service situation.

**Tags:** 802.1D, blocked port, certification, comptia, designated port, root port, security, spanning tree protocol, stp

**Category:** CompTIA Security+ SY0-401

## **Network Separation – CompTIA Security+ SY0-401: 1.2**

One of the fundamental best practices of network security is to segment the network to prevent access and protect resources. In this video, you'll learn about network separation and how organizations can use different segmentation strategies in their infrastructure.

If your organization deals with very sensitive data, you may have not only a requirement to logically separate out the networks on VLANs, but to really create separate physical networks. A physical switch, a physical router, that is completely separated from the other components within your organization.

And when this happens, there's no overlap. You're not connecting them together. You're not somehow creating VLANs between the two. You really are separating them out. This is nice if you're wanting to really, completely separate that data, because there's no way you could get into a switch and somehow end up on the private network.

There's no way that you could reconfigure a router, and somehow end up on that private network. It would really be a separate physical network. When you're in a very sensitive environment— with customer data, with credit card numbers, with health care information— you may definitely want to consider setting up a physical network, and separate it out.

If you don't have the ability to do it with physical devices, then maybe your best option is to set it up with **virtual LANs**, or virtual router-type configurations, or even virtual firewalls. This keeps your costs down.

One of the things that it allows you to do is have these really separated out. The way our technology works these days is, when you virtualize a switch with a VLAN, or you virtualize a firewall, you really are setting up completely separate components. They're designed not to communicate with each other. And that gives you a lot of flexibility, if you need to protect very, very sensitive data from other parts of the organization.

It's an interesting idea, isn't it— a virtual firewall. Here's how a firewall might look. You have a firewall with a lot of ports on the front of it.

One of the things that you can do is, assign certain ports that would be independent from anything else that's inside the firewall. Those ports would have their own firewall rules. Those ports would have their own log files. Those ports would have their own reports. Those ports would be completely independent, and you have administration capabilities on this.

You might want to assign ports 1, 2, and 3 to be, kind of, a red network. They'd be their own separate virtual firewall. Then you'd like to, essentially, build a new firewall inside of that, for these other two ports on here.

These ports 1, 2, and 3 cannot see ports 7 and 8, and vice versa. They are completely separated within the firewall. These are virtual firewall systems. And then you can take some other ports, and put them on a third virtual firewall.

You would log into your firewall, and you would only see your firewall. Even though it's on one physical device, each one of these ports has been administratively assigned so that they cannot talk to each other, they cannot see each other.

If you're trying to save money and, at a very basic level, really separate out your network, that's a great way to do it.

**Tags:** certification, comptia, logical, network  
segmentation, physical, security, separation

**Category:** CompTIA Security+ SY0-401

## **Log Analysis – CompTIA Security+ SY0-401: 1.2**

The event and access logs from network devices can provide a wealth of information. In this video, you'll learn how post-event analysis and real-time analysis of logs can provide valuable security information.

There are a number of universal truths in network security, and one of those is that you're going to have a lot of log files. You're going to have log files from your switches, from your routers, from your firewalls, from your **IPS systems**, from your proxy server, from your **URL filtering devices**. Every device you have on your network has a bunch of log files associated with it.

As a security professional, you want to have these log files. There's an amazing amount of intelligence, and certainly a lot of history that you may have to go back and reference, in those log files.

One of the challenges we have as security professionals then, is keeping it all straight. Usually, you want some way to analyze these log files, without you having to pore through pages, and pages, and pages of logs. There'd be no way a human being could ever read through all of those logs.

So there are systems in place. That you can get to analyze the logs for you. This happens to be a chart that was created from one of those, called **Splunk**, that is designed to take a lot of different log files, consolidate them together, and allow you to put these in a human-readable form, so that you can really see what's going on.

These log files are incredibly useful. If there happens to be a breach, or something that happens, and you'd like to go back in time and understand– what happened during that time frame, what flow of traffic was allowed through, what firewall rule allowed that bad guy to get to our web server. This would be a great use of having all of those logs in one place.

In real time, it becomes a little more difficult. You can imagine the huge amounts of log files that are streaming into these devices. Being able to do any type of real-time analysis of those logs is a pretty complex thing to have happen.

There are tools out there, however, that can parse these logs, and try to keep track of things, at least in near-real time. And sometimes that that's very useful. If you'd like to be able to be identified and alerted as quickly as possible, should something odd be happening, sometimes the only way to know that is if you have something automatically going through all of those log files.

The real key, if you ever get into dealing with tons of log files as a security professional, is just– find a way to automate it. One of the worst things you can do is have all those log files, and not be able to take advantage of them.

So you want to really think about– what would you like to do with the log files? What type of information would you like to glean out of the log files? Do you want to have a real-time view of the world in those log files? You answer those questions first, and you'll be in a better position to understand the type of system you want to have in place to collect the data, and then provide you with some analysis.

**Tags:** certification, comptia, event logs, log analysis, real-time analysis, security

**Category:** CompTIA Security+ SY0-401

### **DMZ – CompTIA Security+ SY0-401: 1.3**

A common network design is to include a **virtual DMZ (demilitarized zone)** to separate the Internet from the inside of the network. In this video, you'll learn how a **DMZ** is used as a layer of protection and which devices you would commonly find in a **DMZ**.

Connecting someone to the internet is something you really have to think about before you go through the process. You don't want to connect someone directly to the internet. There's way too many bad things going on on the other side of your firewall to consider plugging directly in. Unfortunately, a lot of people who have cable modems or **DSL** connections do sometimes connect directly to those links without any type of router or security device in the middle.

In this diagram, I've put down what most people are doing for security when they want to have internet connectivity, and they take advantage of something called a **DMZ**. It's a military term. It stands for **demilitarized zone**, and it stands for that section between the two opposing forces. Usually, it's an area that has been set aside that everyone agrees that nobody's going there to create any trouble.

And in our particular case, we create this **DMZ** with a connection off of a firewall. In the **DMZ** is where you would put things that people need to access from the internet. These might be web servers. They might be email servers. They might be other types of services you're providing to people who are out on the internet side.

And obviously, since your internal network is also connecting through this firewall, generally, people on the inside do have limited access to the DMZ, although sometimes they have no access to the DMZ. It's completely up to you and your security policies.

The idea is that if there is going to be a problem with people accessing resources directly from the internet into your environment, the worst thing they'd ever be able to mess up inside of your network are things in this DMZ. It also means that we can keep some very tight policies on our firewall that allow just the right amount of access into the DMZ and no more. We're not going to give anybody more access to a mail server, or more access to a web server, than that particular service.

And because we have a completely separate network on the inside, the firewall can have very, very tight restrictions on what people can do from the internet to the internal. In many cases, you may not initiate a session from the internet to any device that is on the internal network.

So that **DMZ** is a good middle ground. It's not the tightest and most secure network. It's not completely open to the internet. It's just open enough to provide those services that are important for your internet users.

**Tags:** [certification](#), [comptia](#), [demilitarized zone](#), [dmz](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Subnetting the Network – CompTIA Security+ SY0-401: 1.3**

One of the key concepts of **TCP/IP networks** is subnetting. In this video, you'll learn why we subnet our networks and you'll see a network design that incorporates separate IP subnets.

When you're looking at the network design from a networking perspective, you're trying to maximize the number of people that you can have on a subnet. You want to get the right subnet masking. You're concerned about routing tables within a router.

But from a security perspective, you really want to limit access to resources in the network to only the people who need access to those resources, or you need to make sure that a particular section of the network is more secure than another section of the network.

That's a perfectly good reason, and absolutely a valid reason from a security perspective, to subnet the network— have a segmentation of different devices out there— and then use our router, or even better, a firewall, to route between those different sections of the network. And each one of those subnets would be a world unto itself. And if you ever needed to leave that subnet for some reason, we're going to make sure we examine that traffic and make sure nothing odd is going on.

You're essentially creating a bit of a barrier between the trusted devices on a subnet and something that may be outside the subnet. Now, obviously, there may be subnets that everybody has to talk to. You may have a subnet for your servers.

You may have a subnet for your internal mail servers, for instance. Everybody has to get their mail. So all of those users will be going through your firewall to be able to access that mail server. It's a natural way to keep everybody on their local subnets, but still allow access to those resources that they need.

We might also think about grouping these resources together. The HR department might be on one subnet. The shipping and receiving department might be on another subnet. You might have your executive team on their own subnet. And that way, you can keep any resources or devices that are local to those teams on that subnet. They're local. There's no restrictions to that particular resource. And they have direct access to it. Usually, you get better performance that way as well.

When you're thinking about setting up your subnets on your network, you're really doing it in different ways. You usually have a router right in the middle, or a firewall that is providing a Layer 3 routing function, and it is separating out your network. And everybody who's flowing through the network has to go through that single device.

In many cases, there are multiple devices there for redundancy because as you can see in this diagram, if this is our home office here, everybody has to go through this link. So it has to be a high performance device. It's a device that has to have redundancy. But as long as we can connect all of our subnets this way, we can make sure that they're secure. We can make sure that there's control. And we can make sure that everybody has access to the resources they need.

The subnets also— out on the internet side, there's going to be other subnets out there. And since they're coming through our internet link, that's just another subnet to the world that we'll be able to set policies on and make sure we have the right security set up to the traffic that's going in and out of the internet.

**Tags:** certification, comptia, network, security, subnet

**Category:** CompTIA Security+ SY0-401

### **VLANs – CompTIA Security+ SY0-401: 1.3**

Nearly every network requires the administration of **virtual LANs**, or **VLANs**. In this video, you'll learn about **VLANs** and how they can be used to logically segment and secure your network.

We talked in an earlier video about how useful VLANs can be for providing that segmentation, and this is a very good example of using capabilities within our existing switches to provide additional security. We can go to our switches, which may have hundreds of ports on them, and assign certain ports to certain VLANs. That assignment is something that could be done manually, or it could be integrated as part of our **Network Access Control implementation**.

The way that most people will do this is they might have single switches, and that entire workgroup switch, for instance, may contain an entire VLAN. So, you might have the red VLAN. You might have the green VLAN and the blue VLAN all on a single switch.

But our switches these days provide us with more flexibility. In fact, you might have HR people in one building, and another building, and a third. There might be a shipping and receiving department in all three of those buildings. We can't always have people physically located all in the same place.

So, we do have a capability in our switches, called **trunking**, that will take multiple VLANs and trunk them up to a central switch and be able to route between different VLANs so that you can have a green and a red VLAN plugged into one switch.

Those two devices don't see each other on the same VLAN. In fact, the only way they could communicate to each other is for the red device to go all the way up to the trunk, to the switch— or, in this case, to the router— and come back down to green to be able to communicate to that device. And this could be a firewall, and we can set policies on it. We have **ACLs** that maybe allow or disallow access from the red network to the green network.

And you can then put people wherever they happen to be. They can be in different buildings. We can have people on the red VLAN in this building and this building, people in the green VLAN on our third building and our first building.

This gives us a lot of flexibility, and if we're very, very concerned about what access people have to what resources, being able to segment the network and implement a VLAN technology within our switches provides us with a lot of control.

**Tags:** certification, comptia, security, virtual LAN, vlan

**Category:** CompTIA Security+ SY0-401

### **Network Address Translation – CompTIA Security+ SY0-401: 1.3**

Nearly every Internet-connected organization uses some type of network address translation. In this video, you'll learn how destination **NAT** and source **NAT** is used to provide Internet connectivity and to protect your internal servers from users on the Internet.

A challenge that we all have when connecting to the internet, especially from a networking perspective, is we just don't have enough **IP addresses** to be able to directly connect everyone to the internet, and what we've done is provide network address translation to solve that problem. We have hundreds or thousands of people on our network, all with private IP addressing, but when they access the internet, they're all accessing it through a series of IP addresses that we, as security people, would probably administer.

This is a good example of this. We have a lot of people that are inside of our network on this 192.168 network. Here's 3.22. Here's 1.221 and 1.3. These guys are on one subnet. These guys are on another subnet. When they talk to the internet, they're not going to talk by 192.168.3.22. They actually— you're going to Google or something that's out here on the internet side. They'll send a request to Google, but before it leaves your facility, usually a router— or, in most cases, a firewall— is going to be doing a network address translation.

This is a source network address translation because it's changing the source IP address that's being transmitted out to the internet. So, it's taking and saying you're no longer 192.168.3.22. To the rest of the internet, you're going to look as if you are 66.20.1.12. And when Google receives this request from this user, it sees that the source was 66.20.1.12, and it sends the response back to 66.20.1.12.

It's the job of this device that's providing that translation to keep track of who has been translated to where. So, when that response comes back from Google, that firewall looks through its table and says, now, who did I NAT out to Google to begin with? Oh, I NATed 192.168.3.22. Let me change it back to its original address and send it on its way.

It sounds like that would be a very involved process— that there would be a lot of latency and delay with it. But the reality is it occurs very, very quickly. And to the end users, it's invisible— happens instantly. There's really no delay at all.

Another nice part about this is that we're not connecting our users directly to the internet. There's no way for someone on the internet to access 66.20.1.12 directly and somehow end up at this user. The only way you would be able to do that is if that user asked Google first, and Google replied. So, there is a security component to providing that source NAT.

But what if you do have web servers and you do have email servers, and you would like people on the internet to be able to access those devices directly? Well, you do the NAT in the other direction. You do something called a destination NAT. You configure your firewall or your router to say if you ever see an IP address coming in of 66.20.1.14 and that destination port number, perhaps, is TCP 80, we could say that that should be going to our web server.

So, what we'll do is, if we have an inbound flow and it hits our firewall and it says 66.20.1.14, port 80, well, I'm going to convert that. I'm going to do a destination NAT and change the destination IP to really be 192.168.3.22.

Since this is also usually on a firewall, you can also set security policies to that. You may only do port 80 to that device because that's our web server. Maybe port 80 and TCP port 443— that's the only thing you can do to that device. So, you can set up many different IP addresses that go to many different devices. You can set up one IP address and simply decide what port numbers are transmitted to what devices on the network. That's more of a **PAT**, a **port address translation**.

There's a lot of flexibility that you get with this. But it does provide you with a significant security advantage here because we can now decide who gets to talk into our network, how they get to talk into our network, and what specific devices they can talk to.

**Tags:** certification, comptia, dnat, nat, network address translation, security, snat

**Category:** CompTIA Security+ SY0-401

### **Remote Access – CompTIA Security+ SY0-401: 1.3**

Most of us rely on remote connectivity to enable us to perform our job. In this video, you'll learn the important security considerations for remote access.

We've created all of these security infrastructure devices, and firewalls, and network address translations, and network access controls because we're always concerned about who's connecting to our network.

Well, now, of course, our users are much more mobile. They are in coffee shops. They're in home offices. They're traveling around the world. But they still need access to important resources that are on the inside of the network, and that is one of the primary balancing acts you have as a security professional, is how do you provide the important business access to these devices but still keep everybody secure?

So, remote access becomes a very, very important component of this. Whenever you start looking at how you're going to get people to communicate back securely into your network, you're almost always going to use some type of **encryption technology**. In later videos, we're going to talk about some very specific methods of being able to authenticate and encrypt traffic, especially over a remote access piece.

But obviously, **encryption** is incredibly important because you have no idea who on the internet might have access to this data as it's flowing through. If you're in a hotel, the hotel certainly has access to that. In fact, other people within the hotel who are staying there might have access to your data as well. It's very common on wireless networks for people to sniff the air— be able to find information that you may be sending in the clear. So, creating an encrypted tunnel prevents some of those things from happening.

You might also want to add on additional technologies to provide additional authentication functions— for instance, be able to have a token generator of some kind, whether a hardware token generator or one in software, that's constantly providing these pseudo-random numbers to you.

So, not only do you have to put in your username and your password, but you also have to put in some other piece of information. Usually, it's based on something that you have with you, something like a token generator. You have to now type in username, password, and 778645, and hit Enter.

And these numbers, of course, are updating themselves every 30 seconds, every 60 seconds. So, when your end server sees that this is your username. This is your password. A-ha, that must be you because you happen to know the secret number that was popping up during that last 60-second period— now, just another method, another thing that we can add to make sure that people from outside are really who we think they are.

If you're doing any type of remote access, you should always be looking at your logs. You should always be checking them to determine who's who is connecting from where. And you can also set up methods within your remote access devices to make sure that people aren't logging in from multiple places. If somebody's mobile and you see that they're

logging in from Starbucks in one city and Starbucks in another city, you might need to question that. One person obviously can't be in two places at once.

So, you not only want to look at your logs and see if that's happening, but also set up some security controls within your remote access equipment to make sure something like that doesn't happen.

**Tags:** certification, comptia, remote access, security

**Category:** CompTIA Security+ SY0-401

### **Telephony – CompTIA Security+ SY0-401: 1.3**

**VoIP** and **telephony technologies** are now integrated into almost all of our networks. In this video, you'll learn what security concerns exist for telephony-related technologies.

A technology that is really past the point of an emerging technology— now it's embedded everywhere in everybody's network, it seems— is **Voice over IP** and other telephony-type functions. Now we have our phones using the network to communicate, sometimes communicating to phones that are on the plain old telephone system— the **POTS system**— outside of our facility. Sometimes we are doing complete digital communication between sites, all being done over our network. We no longer have third-party phone lines in a traditional sense to communicate via voice and communicate in other ways through this technology.

The problem is that it is a relatively new technology and it's very difficult to secure. It's a very complex technology. It's not simply transferring a file. There are control protocols. There's a protocol for when you're picking up the phone and dialing. There's another set of protocols when you're sending voice communication or video communication over those links. So of course you have to check every bit of this every step along the way.

And you have security concerns of people being able to get into your voice systems. You have security concerns of people denying access to these voice systems, make it so you can't use your telephones. So you really want to have firewalls and other security technologies in place.

But every Voice over IP and telephony system is a little bit different. And because of the way that they embed the IP addresses inside of some of these Voice over IP protocols, simple firewalling of port numbers isn't necessarily going to work well for you. You usually use something called an **application gateway**. In fact, it would be a **real protocol-specific** or **phone-specific application gateway** that understands that Voice over IP technology that you're using, and is able to communicate properly. It's able to do NATing properly. It's able to firewall it correctly and send it through encrypted tunnels the way that it should.

Usually this is something that the provider of the firewall or the application gateway makes you aware. Oh, this works fine with this manufacturer's telephones. So if you're implementing Voice over IP and you need to secure it— you need to make sure that your security technologies know about that technology, especially that very specific manufacturer of that telephony technology and it's able to handle it properly.

And of course, don't forget your other phones. Just because Voice over IP is out there doesn't mean you have gotten rid completely of maybe your old phone system. So if there are older phone systems in place, make sure that those are secured properly as well. Make sure that people are not able to use those and make long-distance calls over your system. I shouldn't be able to walk into a conference room, pick up a phone and start costing you money. So don't forget about your old technologies. And of course, find the security features in the new technologies that you need and make both of them all work together.

## **Network Access Control – CompTIA Security+ SY0-401: 1.3**

An open network jack in a conference room can provide a malicious user with complete access to the internal network. In this video, you'll learn how **network access control (NAC)** can be used to help secure network connections.

In a previous video, we talked about 802.1X, we talked about network access controls, we went through how the conversation starts when you're doing network access control with 802.1X on the network, and, as you've probably seen, it's not the simplest thing to get running. There are a lot of different components, a lot of moving parts. Once it's in place though, it runs extremely well. It's a lot of work to get that there and, although it is very complex, you put it in right, it's really going to work well for you.

You most often see this in very, very large environments. You're not going to run into a lot of network access control requirements if you're in a building of 10 people and you don't have to worry so much in that small office of who's got access to the network. You can see, they're sitting at their desk and they have access to the network. But, in universities, in large enterprises, in environments that are very diverse, you have people in many, many different locations, geographically spread out, it makes a lot of sense to think about some method of network access control.

But you're going to need all of the components to make it work. You're going to need the servers for authentication. You're going to need the databases. You're going to need a way to maintain those databases with those username and passwords, maybe integrate into an existing name services that people are using for other authentication methods. All of those systems will need to be redundant because if one goes down, one breaks, then nobody would be able to log into the network.

So all of these different things have to be thought about from a security perspective before you go into doing any large, widespread use of network access control. In fact, it's an entire section of the Security+ exams. Section 5.0 is Access Control and Identity Management. The entire section is dedicated to understanding and learning more about network access control. So, if you have any questions about how you're going to implement those things, why it's important, and how we're going to address that and the Security+ exam, you can go to section 5.0 and go through all of those videos.

**Tags:** [802.1x](#), [certification](#), [compTIA](#), [nac](#), [network access control](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Virtualization – CompTIA Security+ SY0-401: 1.3**

The virtualization of the data center is now a standard part of our networks. In this video, you'll learn about the security concerns that involve virtualization technologies.

In our previous video, we talked about the configuration of how people might virtualize a firewall and the benefit there obviously is there's a huge cost savings. If you don't have to buy five firewalls, you could buy a single physical firewall and simply split it up into five virtual pieces. Might save a lot of money there. We're seeing virtualization on file servers. We're seeing virtualization on huge systems that are put in place. You buy one big monster server that now can hold 100 or more different servers inside of it; there's some obvious cost savings there. There's cost savings on the purchase of the hardware. There's cost savings of the place you would put it into your data center, the cooling systems, the maintenance of the system, so there's a lot of advantage there.

But obviously, there's security issues when you think about these things as well. You now have one device with 100's or more servers inside of it. How do you protect information that is transferred between those servers? You can't touch them. You can't plug into the network between them. They're all self-contained in a single unit. You don't have that

ability to touch a physical object anymore. So we're having to change the way we think about security and virtualization. We have to think about how we would get virtual servers and virtual security devices into the virtual environment. Or we're putting physical security devices outside of the environment and forcing those servers to communicate through our security components. There's advantages and disadvantages to both ways.

So you really have to think about how you're going to do this. How you're going to implement it. It is an emerging technology and security to create virtual firewalls that work on some of these virtual servers. It's something that, as we continue down the road of virtualization and this gets more and more mature, I think you'll start to see more intelligent technology come out from a security perspective. In the meantime, you've got your logs. You know exactly who's going where. You know what files are being accessed. You know what resources are being used. We talked in an earlier video about how important it is to have all of those great log consolidation tools available. So you can really look through all of those logs in one central place and see everything going on. So for virtualization, try addressing the security in many different places. You're going to have to really work with this technology to figure out how you're going to keep track and maintain control over all of those virtual resources.

**Tags:** [certification](#), [comptia](#), [security](#), [virtualization](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Cloud Computing – CompTIA Security+ SY0-401: 1.3**

The term “**cloud computing**” has many different meanings, depending on the context. In this video, you'll learn about platforms as a service, software as a service, infrastructure as a service, and the different cloud deployment models.

The term **Cloud Computing** has been casually thrown around by marketing professionals, but it's actually a very good way of deploying applications in ways that are very flexible. In this video, we'll look at a number of different ways that cloud computing can be used, and I think after watching this video, you'll find you're already taking advantage of resources and applications that are using these cloud computing infrastructures.

The first implementation of cloud computing that we'll look at is called platform as a service. You may see this abbreviated as **PaaS**. In **platform as a service**, you don't have any software. You don't have any hardware. You don't maintain a data center. There's no heating and there's no cooling. You are just the end user. And you're taking advantage of someone else's infrastructure and simply using the platform that they've created to run your application or take advantage of a service.

The challenge from a security perspective though is that you do not have direct control over any of this infrastructure. You can't touch the application. You don't have control over the security of a server. You don't handle the server patching. So you may not have a direct security control over when those types of patches are applied. There is a completely different group of trained professionals that are in charge of keeping that service running. And they're going to make sure that you have access to this platform, and you're able to use the resources on this platform. But everything behind the scenes is something that is hands-off from you directly.

A good example of platform as a service is **salesforce.com**. **Salesforce.com** is a **customer relationship manager**, or **CRM**. And they provide a very flexible front end that allows you as the end user to customize exactly how you would like to use their platform. On the back end are the servers and the databases, but generally speaking, you don't even have to worry about that part. In fact, you hardly ever see that part of the platform.

This allows every single customer to create an experience that's specific for them. And they don't have to worry at all of the platform that's providing that service on the back end.

In the past, if you wanted to provide payroll services for your organization, you would need to go to a third-party. You would purchase their software application. You would bring it inhouse and generally install it on your own servers. And then you would be able to create the payroll and process that payroll every month.

With software as a service, we've taken that entire process and made it completely turnkey. In fact, the software is usually posted somewhere else through a third-party. And all we simply do is log on to their services to be able to perform that particular tasks. So this way you wouldn't necessarily need to run your own mail server inside of your organization, or even have your own accounting department with their own platform to be able to provide that payroll service. All you would need to do is log into this software as a service and use the software hosted through a third-party and managed through a third-party to perform these particular tasks.

From a security perspective, using software as a service is very different than having your own servers running your own software in your organization. Take for example payroll. If we use software as a service, we're connecting to a third-party and using their resources, and putting all of our payroll information on that centralized database that's somewhere else. Of course, that database is ideally private to us, but it is something that's now stored outside of our organization. So we have to think about what type of data we're putting into the cloud, and if someone was to gain access to that information, how would that affect our organization.

A good example of software as a service is something like **Google Mail**, or any of the other hosted mail services. We're not running our own email client. We don't have our own email server. We don't have to maintain the mail exchange information in our **DNS**. All of that is handled separately through this software that's running on someone else's computers. And of course, it keeps all of our information private. We authenticate into this cloud, to the software as a service on this mail server, but of course we're always concerned about someone else also authenticating as us and gaining access to that data.

Another implementation of cloud computing that you may see is infrastructure as a service. You may hear this also called hardware as a service, because we are simply acquiring hardware that we could use for our own software. In fact, this hardware may not even have an operating system on it. We are simply taking advantage of hardware that may be located in one or multiple places anywhere in the world. From a security perspective of course, we're still responsible for this. In fact, we're even more responsible for this, because now we are in charge of securing the operating system. We're in charge of securing the software that is running on this hardware. The data is still out in the cloud. It's outside of our organization. So we have to be very careful about how we implement security on this hardware that we are acquiring in the cloud.

You might see infrastructure as a service used if you ever want to build your own web server, but you didn't want to do it on a shared resource. You wanted your own hardware that you would run your own software on. In fact, you would have complete control of the operating system. You might also see this if you were hosting an email service externally. And you had your own software for email, and you wanted to control that software, and you just need hardware that was located in the cloud to be able to run that.

And of course, this also allows you to very easily scale, because all of the hardware is located somewhere else. You're not having to purchase new hardware. You simply buy the hardware in the cloud and load your software on it as you need to expand capacity.

Up to this point we've always talked about the cloud as being something that's outside of our organization, external, and we don't really have control over it. But the reality is that we could build a cloud anywhere, including in our private data center. And it's very common these days to see a private cloud that we can then pull our own servers out of

the cloud and deploy our own infrastructure as a service, or deploy our own platforms as a service internally within our organization.

You generally see this with larger organizations that have multiple data centers, but it can be done in any type of environment. The kind that we usually talk about when we refer to cloud computing is generally the public cloud, where everyone has access to these resources that are located anywhere in the world. And occasionally there might be a mix of these— a hybrid of public and private. It depends on how your application is used in your environment. You might want to keep your data local, but have the platform as a service located externally in the public cloud.

You may also see a community model of cloud computing, where there might be a central resource in the cloud, like a mail server, and multiple organizations are using that exact same resource to be able to use that service. Something that allows the cloud provider to scale up very easily and support many different customers all on the same platform.

**Tags:** certification, cloud  
computing, community, comptia, hybrid, iaas, paas, private, public, saas, security

**Category:** CompTIA Security+ SY0-401

### **Defense in Depth – CompTIA Security+ SY0-401: 1.3**

One of the fundamental foundations of information security is the concept of defense in depth. In this video, you'll learn about defense in depth and some of the technologies used to implement defense in depth.

When you're planning a security strategy for your organization, you never want to rely on any one thing. You want to have security throughout the organization protecting you at every step along the way. This layered security is something we call in the industry defense in depth.

Let's look at a number of different layers that we can add to this defense in depth. This is in no way a comprehensive list. But it should give you a pretty good idea of what you could put in your environment.

Obviously, a firewall comes to mind, something that we can put at the edge of the organization as people are going in and out to the internet. Many organizations will put their firewall on the inside, either protecting different floors of a building from each other and, in some occasions, protecting different devices inside the data center. It's a good way to make sure that only the data flows that you want are going to be running inside of your network.

Another methodology that is really associated with many firewalls is **implementing a demilitarized zone**, or a **DMZ**. A **DMZ** is a bit of a middle ground between the inside of your network and the outside so that people who need to access resources in your organization don't come all the way to the internal parts of your network. They go into this middle ground called the **DMZ**. And it's just another way to have that layering of security in your environment.

We often take authentication for granted as a security strategy. We obviously need to authenticate people. We usually do this by using a username and a password.

But, of course, you can have multifactor authentication that might layer on additional security. You could require someone provide the random or pseudo-random message from a separate multifactor authentication key. Maybe you would provide someone with cards that they would use to be able to provide the additional factors. But every single one of those adds another layer to the defense in depth.

If you can add an intrusion prevention system to your defense in depth that would be able to watch all of the traffic that's traversing the network, and if somebody is trying to take

advantage of a vulnerability on a server or workstation, this intrusion prevention system would be able to stop those traffic flows. It's another way to sit behind the scenes, watch the traffic go by, and only stop the bad things while you're allowing all of the good traffic to proceed.

If you're connecting in to your network from the outside, then you're probably using a **virtual private network**, or **VPN**. The **VPN access** encrypts all of the data as it's going through the internet. And only once it gets inside of your network is that traffic decrypted and sent on its way in the clear.

This means that you could be in a relatively insecure location, like a coffee house. You could be a hotel where the wireless network is connecting you in the clear. But all of your traffic is going through this encrypted tunnel. So although the wireless network is one that people could access and listen to your data going by, your information is still secure as it's passing through that insecure network.

We've become very accustomed to running anti-virus and anti-malware software on all of our workstations. And that makes sense because that's the last possible place where you could stop the malicious software from executing on your workstation and infecting your machine. Another step in that layered defense-in-depth security strategy.

And as I mentioned earlier, this is just the beginning of a defense-in-depth strategy. In fact, it's not unusual for medium to large organizations to be running every single one of these technologies in their environment. It's one of those things that you'll notice as you go through the Security Plus certification that you'll start to see other strategies that you can implement to provide defense in depth in your organization.

**Tags:** certification, comptia, defense in depth, security

**Category:** CompTIA Security+ SY0-401

## **IPv4 and IPv6 – CompTIA Security+ SY0-401: 1.4**

**IPv4** and **IPv6** are some of the fundamental building-blocks of network communication. In this video, you'll learn the structure and design of the **IPv4** and **IPv6 protocols**.

Let's start with some of the most common protocols you'll find, **IPv4** and the **IPv6**. And you can see if you're using a computer today, you're probably using **IPv4**. This is the most popular protocol known to man, I think.

This is **Internet Protocol version 4**. These are addresses that are **32 bits long**. And because they're 32 bits long, if we do the math, there's a maximum number of **4,294,967,296 addresses** you can get. Well, as we've probably seen in the trade magazines in the news, that is not a lot of addresses. We are running out of addresses.

So we needed something bigger and badder, something that would really give us flexibility at least into even the unforeseeable future, to provide us with more capabilities. And so we came up with a new version of **IP**, **TCP/IP version 6**. These are **128-bit addresses**. And these are much larger. You can see the total number of addresses available now, which means that you can have everybody in the world with many, many, many of their own addresses and then some.

You can cover the entire world with IP addresses now. Parts of the ocean can just be covered with **TCP/IP**. And we'd still have plenty more left over for everybody else. So a lot of flexibility, a lot of capabilities there. And we'll talk a little bit about some of the additional advantages that **IPv6** is going to bring to us.

You'll notice also that we moved between **IPv4**, we just jumped right to **IPv6**. And that's because in the standards realm, there was something called an **IPv5**. It was called the **Internet Stream Protocol**, or **ST**. It obviously never really took off. But it is one, because that name was taken, we couldn't really repeat it. So going to **IPv6** was the next step. And that's why you have that jump between the version 4 and the version 6.

**IPv4 addresses**, as we said, are **32-bit addresses**. But we almost always represent them in decimal form, in a base 10 form. And we call this a dotted decimal form, because you have four decimal numbers here, 192.168.1.131. And it's got these dots between. It's got a period between all of them.

Now, behind the scenes the computer obviously isn't really representing these in decimal numbers. That's for us human beings. It's representing them in binary. And here is the binary representation of each of these what we call octets, this group of eight all stuck together.

You can see 192 is 11000000. 168 is 10101000. So if you put all those together, it comes out to be 168 whenever you're converting that between binary and decimal.

You might hear these different sections referred to as an octet, because it's eight of those bits put together. **Eight bits** put together is also a byte, at least it is in the computing architectures that we tend to use. And so whenever you hear these different bits or bytes or octets, that's what it's referring to in this dotted decimal notation for **IPv4**. And that's where we get the 32 bits of an address is when you add all of that up, that's how big they are. So if each one of these is eight bits long, if all of these were turned into a 1, the maximum number that you could ever have for any section of these dotted decimal **IPv4 addresses is 255**. And that is how we're typing in our IP addresses into our systems, into our firewalls. And we're deciding where people can go to what subnets are based on usually these IPv4 addresses.

IPv6 addresses are much bigger. There are much more things to look at here. In fact, we've got away from dotted decimal. It's now we're separating out and representing an IPv6 address in hexadecimal. And that 16, that base 16 functionality provides us with addresses that look like, Fe80, a couple of colons, 5d18, colon, 652, colon. The colon is now really taking the place of that decimal that we were using before.

And you'll notice in some cases we have two colons stuck together. That represents a series of zeroes stuck between. A real IPv6 address if you're going to write it all the way out would look something like this.

These sections of four, it's really these two hex values that are all stuck together. And they have colons that are separating them. But fortunately, we've created some shortcuts for us that if there's a string of zeroes, we'll just put two colons in place and assume that those are zeroes between them. If there are leading zeroes in front of a couple of numbers, you can always get rid of the zero. So it helps a little bit.

And of course, if you were going to write it all out in binary, you can see the representation of all the binary numbers all spelled out on the screen for us. Now, this is now a lot of data going by. You can see those are 128 bits or 16 bytes for a single IPv6 address. So there's a lot going on there. And as you can imagine, typing in the IP address, the IPv6 address of a web server or the IPv6 address of your email server isn't really going to be an easy thing any more when we move to IPv6. So your domain name services are going to be extremely important, being able to type in mail.google.com is going to be a lot easier than typing in a big, long IPv6 address.

**Tags:** [certification](#), [comptia](#), [ipv4](#), [ipv6](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **IPsec – CompTIA Security+ SY0-401: 1.4**

If you're concerned about the security of your network connections, then the **IPsec protocol** suite is for you. In this video, you'll learn the basics of **IPsec technology**.

**IPsec** is a very, very common protocol to see on the internet. It's one that encrypts at **OSI** layer three. It encrypts at the IP layer, which makes sense, it's called **IPsec, IP security**. Whatever's inside of that IP doesn't really matter. It's irrelevant because we're encrypting right there at the IP level. We can put anything inside of that. We can put any type of **TCP** or **UDP** or application, but the reality is **IPsec** doesn't care. It's going to encrypt it all up and it's going to send it off via that layer three communication.

It is an open standard. It is used in a lot of places. It's used for **VPN connectivity** from a client to a server. It's used between firewalls to be able to encrypt data between those devices. You'll see IPsec a lot if you get into security or doing a lot as a security professional, you'll use IPsec every day to do a lot of different things. This is usually something where you're talking about bringing up a tunnel. You're creating an encrypted link between devices and IPsec is going to provide authentication. It's going to make sure that the data is getting to and from where it's being sent in one piece, and nobody's changing it, with integrity. Confidentiality is there in encryption so we can make sure that nobody's able to tap into this link and see what's inside of those connections.

A very, very useful technology, a very robust technology, one that's used in many, many different places. And because it is a standard you can get Manufacturer A and Manufacturer B's device to communicate to each other via IPsec. Everything's encrypted and yet the devices were made by different people. A lot of advantages there and if you're going to be doing a lot with networking and security, you'll certainly run into IPsec.

**Tags:** [certification](#), [comptia](#), [IPsec](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **ICMP and SNMP – CompTIA Security+ SY0-401: 1.4**

Every network and security manager makes extensive use of network management protocols. In this video, you'll learn how **ICMP** and **SNMP** can be used to help manage and protect your network.

A useful protocol that's used all the time by network administrators is **ICMP**. Stands for **Internet Control Message Protocol**. And it's often used to be able to send little messages between computers. An echo request and an echo reply are a couple of very common examples of this. You type in ping and an IP address, what really is happening is an **ICP request** is made to another server and hopefully you're receiving an **ICP reply**. We call those a ping request and a ping reply, but behind the scenes, that's exactly what's happening. Now this is really useful for troubleshooting.

It's very useful for bad guys to be able to do reconnaissance of your network as well. Imagine being able to ping all of the systems in a particular subnet and find out who responds back. Sometimes that's not what you want. That's too much information for the bad guys. And you'll notice that most firewalls, one of their default firewall configurations is to disallow **ICMP** packets to go through the firewall.

Now, there are other methods of ICMP that can do things like redirect different network connections. You could tell a router that that network's no longer located over here, it's located over in somewhere else. That might be a legitimate message or that may be a completely illegitimate message. That might be the bad guys trying to redirect traffic over to their machines. So, usually those are also restricted– at least those particular **ICP types** of messages may be restricted. And if you ever go out somewhere and you see that you're trying to ping a device and you're trying to ping it says, sorry the destination host you're trying to ping is unreachable, that's because either that device really isn't on

the network or our security administrator was very smart and restricted access of ICMP through the firewall. No matter how much you try, you're never going to get a response back.

Another useful network management protocol is **SNMP**, that stands for **Simple Network Management Protocol**. It's used between devices from a management perspective to gather details, metrics, about how those devices are performing. Very often, you'll have a device that's in charge has some **SNMP software** on it that is simply querying the devices, usually infrastructure devices in your environment, maybe asking a router, "hey, on a particular interface, how many bytes have you seen come into that interface?" And the router responds back with 210,506 bytes. Kind of boring. Kind of very dry, but if you start compiling those metrics over long periods of time, you can start to see how much bandwidth might be going through a router.

And these **SNMP queries** that can be done across many different variables. Many different routers will have hundreds or thousands of **SNMP variables** that you could query. What's the temperature inside of a server? How many people are connected to the device at this time? And you can gather a lot of details. If you're using SNMP Version 1, there's different versions, three different versions of SNMP currently, this is the one that was original and you're really asking for some very specific information and getting a response and that entire communication is completely in the clear. It is not encrypted, it is not verified, and anybody who happened to be on the network between those two devices can see exactly what you're doing.

Now in Version 2, we added some additional capabilities. We could do some data type enhancements of what we were asking for. We can ask for many different metrics at one time and get a big bulk response back. But there was still no encryption. There's still, from a security perspective, a real concern there.

So, **SNMP Version 3**, which is the latest standard, it's the one from a security perspective, you should almost always be insisting on because it checks the integrity of the message that it really came from that device, was authenticated to that device, the message wasn't changed along the way. So that integrity and authentication become very, very important. And it's encrypted. And the data that's going between the SNMP device and the reply that we're getting back is something that nobody can tap into, to really see what's going on.

And because this SNMP information can be very detailed about configurations, and status, and amount of traffic, and a lot of details, you need to make sure that only you or people who are specific to understanding how those devices are running, maybe the manager of those devices, can only see this data. This is very, very critical data. If you know what to look at with SNMP, you can gather a lot of information about those devices.

Also from a security perspective, you should make sure that the devices that are out there that support SNMP can only be queried by your SNMP devices that are out here doing the querying. You don't want a third party to plug into your network and instantly be able to gather information via SNMP. So that's an important security concern, especially if you're using SNMP with all of your devices in your environment, to check on the status and availability of those machines.

**Tags:** [certification](#), [comptia](#), [icmp](#), [management](#), [security](#), [snmp](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Telnet and SSH – CompTIA Security+ SY0-401: 1.4**

**Terminal access to remote devices** is an important part of network and security management. In this video, you'll learn how to use **Telnet** and **SSH** to manage your network infrastructure.

As a security administrator, you will very often be called on to configure the security devices that you may have located all over your network. These may be located in the same building that you reside in or they might be in a different country. So in those cases, you need to be able to sit at your desk and then remotely access those devices. Two of the very common ways to do this is through a **Telnet console or an SSH console**.

**Telnet** stands for **Telecommunication Network**. And it's a very common way to be able to connect to a device remotely across the network. This gives you a console view, very much like the one that you see here. Very much text based, usually there is a command line involved. The important thing to remember with Telnet is that all of the communication between your workstation and that telnet service is going to be completely in the clear. None of this information is encrypted, including your username and your password that you use to log on to this device. That's obviously an important security consideration. So if you are in a production network and there is a lot of security that you're concerned about, Telnet may not be the best way to connect to these remote devices.

If you're running on a Linux workstation or an OS 10 workstation, you may find that the Telnet client application that we're about to use is installed by default. If you're on Windows, the Telnet client is not installed by default in the Windows configuration. You'll have to go back into the Windows setup area to add that particular component to the operating system. And that should give you a pretty good idea that using Telnet probably is not the first place you should go if it's not included in the operating system by default.

In fact, if you're running in any type of secure environment, in any size organization, you're probably never going to use Telnet. But there are a number of Telnet servers out on the internet. Let's try connecting to one. I'll simply type Telnet. And one that I'd like to use is called **rainmaker.weatherundergroundwunderground.com**. This is hosted by the Weather Underground service. It gives you information about weather forecasts. And you could see it even says, "Welcome to the Weather Underground Telnet service." And it says the National Weather Service information. That's what's in here. Press "return" to continue. We will press "return" and it says, "enter a three digit city code." I'll put TLH for Tallahassee. And it gives me the information about Tallahassee: the temperature, the humidity, the winds. And I can use this Telnet front end to gather other pieces of information.

Telnet was a good choice here because it didn't require any type of encryption. There is no username and no password and allows the Weather Underground service to provide this to many different resources. If this was a security device, however, and we were wanting to not only protect our login information, but protect the information that was being sent back to us, then we would probably want to use a protocol like SSH.

**SSH** stands for **Secure Shell**. And this gives us the same console front end, but all of the communication behind the scenes is encrypted. Means everything from your computer to the device you're talking to on the other side is not something that someone could connect to, gather packets, and be able to piece together the information the way back and forth. Obviously, in most environments, this is exactly the type of console that you want to use because you could be assured the nobody's going to pick out your username, password, or any of the communication between you and that other device.

If you're running in **Linux** or you're running in **OS 10**, **SSH** is a natural part of the operating system. It's generally installed in most builds of those operating systems. If you're using

Windows, you may have to use a separate application, such as **PuTTY**, to be able to use an **SSH console**. We use the Telnet console earlier to connect in and look at weather.

Now let's use SSH to do something that's a little more fun. We'll SSH to nethack@alt.org. This is going to alt.org and it's using NetHack as the name that it will use to connect to this device. In the case of this NetHack game, this initial connection doesn't require any particular passwords. It simply presents me with the screen for NetHack. And you can log in, register as a new user, watch games in progress. We'll hit W. Here's a list of the games in progress. Now, notice that the console looks remarkably similar to using a Telnet console, and it should. Both of them look exactly the same. But now all of this communication behind the scenes is completely encrypted.

Usually we'd be using SSH to connect to a firewall, or a router, or a switch, and we'd be making some configuration changes. This information, obviously, is relatively sensitive. So using the SSH protocol insures that all of that communication will remain secure.

**Tags:** [certification](#), [comptia](#), [security](#), [ssh](#), [telnet](#), [terminal](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Transferring Files – CompTIA Security+ SY0-401: 1.4**

There's more than one way to send a file across the network. In this video, you'll learn which file transfer technologies can be used to send in-the-clear and encrypted information.

In an earlier video, we looked at using **Telnet and SSH** to be able to remotely connect to a console of a device. But sometimes we need to transfer files back and fourth to that device. And there's a number of protocols and methods that we can use to do that. One method for transferring files that's been around for a very long time is one called **FTP**. That stands for **File Transfer Protocol**. It's very common you'll find an **FTP service** available on many different devices. And there are many **FTP clients** even built into most operating systems. One important consideration of FTP, however, is that this entire communication process is in the clear. In much the same way that Telnet allows you to communicate to a remote device console and have in-the-clear communication, FTP works the same way.

When I FTP to a device, I can see the user name and the password very clearly written in the packets going over the network. So if you want to maintain security of your systems, you will probably not be using FTP to transfer this information.

For more secure file transfer type, there's a number of different options available to you. One of them is **FTPS**. This stands for **FTP over SSL**. You may see this also abbreviated as **FTP-SSL**. This stands for **File Transfer Protocol Secure**. There are other File Transfer Protocols that add security features to it. Another popular one is **SFTP**, which we'll talk about in a moment. Now keep in mind that **FTPS** is not the same as **SFTP**. Both of these use completely different mechanisms to be able to communicate between the systems. So if you have a system that supports **FTPS**, you cannot connect to it with an **SFTP client**, and vice versa.

The **FTPS** is very commonly seen on web servers because there's an **SSL component** already on that web server and ready to go. On servers that are running SSH, you also tend to see other types of secure file transfer mechanisms like **SCP**. That stands for **Secure Copy**. And it uses SSH as the underlying mechanism to be able to transfer files. **SCP** is a very bare-bones file copy process. It doesn't really allow you to do much with the operating system file structure. It really is designed to transfer a file both to and from a device and that's about it.

If you need more capabilities, then you'd probably want to use something like **SFTP**, which uses the same **SSH** functionality to be able to transfer this. This stands for **SSH file transfer protocol**, but it gives you a lot more functionality to be able to manipulate the file system. So you can resume transfers that may have been interrupted. You can modify directory listings or folders. You can remove files from the system, all using **SFTP**.

In fact, this screenshot here shows you an SFTP session. And you can see the local files. You can see the remote files. And you have a lot of functionality for being able to manage all of your file transfers both on the local machine and the remote device using the SFTP protocol.

**Tags:** [certification](#), [comptia](#), [encrypted](#), [file transfer protocol](#), [ftp](#), [ftps](#), [scp](#), [security](#), [sftp](#)  
**Category:** [CompTIA Security+ SY0-401](#)

### **DNS – CompTIA Security+ SY0-401: 1.4**

We rely on the **DNS protocol** for almost everything that we do on our networks. In this video, you'll get an overview of the domain name services protocol and learn how it is used to convert names to **IP addresses**.

Our **Domain Name Services**, or **DNS servers**, are very important in **IPv4 and especially IPv6**, because their job is to take a name and convert it to an IP address. So if you were to look at and you type into your web browser [www.professormesser.com](http://www.professormesser.com), behind the scenes, your browser asks a Domain Name Server somewhere that it knows about, hey, do you happen to know how to get there? I've no idea the IP address of this website. Do you happen to know what that is? Well, sure. Here's your answer. 74.208.221.234. Obviously these are very, very important resources, very critical resources from a security perspective. First, if somebody takes down your DNS, they could essentially take down people's access to your services. Because I don't have a list of everybody's IP address. I rely on my DNS server for that.

You also have to be careful that nobody gets access to your DNS server and changes the IP addresses in there. If somebody was to do that to my DNS server, they would type in Professor Messer and might end up at a different IP address, perhaps even one looked exactly like my website. And that is phishing. That's a problem. You could ask people to put in their user name and password thinking they were putting into a real website and they were not.

Now that is something that does occur. It's very rare to have somebody break into a DNS server, because security professionals understand how important that resource is and they tend to keep that very, very secure.

You also have to watch out just for plain old redirection. You don't want somebody going to your site and suddenly ending up on a competitor's site because they typed in your particular name and the DNS server was completely wrong. Somebody got in there and changed where people were going.

So **Domain Name Services** is incredibly useful. The way you would perhaps even look some of these up is to use the nslookup command. And if you use the nslookup of [www.professormesser.com](http://www.professormesser.com), it even tells you it goes out to my DNS server, which is 8.8.8.8. That is Google's DNS. And it says, here's the answer. ProfessorMesser.com can be found at 74.208.221.234.

**Tags:** [certification](#), [comptia](#), [dns](#), [domain name services](#), [security](#)  
**Category:** [CompTIA Security+ SY0-401](#)

## **HTTPS and TLS/SSL – CompTIA Security+ SY0-401: 1.4**

Without encryption, we would not be able to securely use our network connections. In this videos, you'll learn how your browser encrypts all of the information you send to a web server using **HTTPS or TLS/SSL**.

In the security world, obviously encryption is extremely important. We want to be able to make sure data that is sent across the internet is something that only I am able to see and the web servers are able to see.

So, for web pages we use a number of different encryption technologies to be able to do this. You'll see this often represented as **HTTPS**, which is our **Hypertext Transfer Protocol with the S on the end and S means secure**, which means that's an encrypted connection. So, what you're doing is essentially setting up that encrypted link to that web server and it's using an encryption method called **TLS**, which is commonly called **SSL**, although that's not technically an accurate representation of what this is. But when somebody says, "I have SSL encryption on my web server," that is what they're referring to. It's able to then do **HTTPS**. Now, this **Transport Layer Security**, and some people still call it **Secure Sockets Layer**, is what's really doing the hardcore encryption for our server.

**SSL** was an encryption technology created by Netscape way back in the day, and it was updated and a standard was created by the **Internet Engineering Task Force**, the **IETF**, that updated it and created a new name for it called **TLS**. So, we can see that that's a little bit different than **SSL**. Now, the reality is that the web servers that you're connecting to are really encrypting the data with **TLS**. Even if they say on the web server, "This is SSL encrypted data," it's really **TLS** that's doing it. And the way you can tell is to go to your browser, and there's a lock on your browser that shows you that the data is encrypted, and that's the method that's being used in your browser to be able to do that.

This technology is also used in things outside of the browser, and you don't have to use **HTTPS** and that **TLS** encryption in the browser, third-party applications can use those as well. It's an encryption technology and something that's very easy to implement because the libraries are open and available for anybody to use. So people will sometimes use this to hide information from the security folks. Because they're using their own devices with their own encryption certificates, you don't have access to be able to see some of those things sometimes. So, if you see a large amount of **SSL** or **TLS** type traffic on your network and you're wondering, "Where's that coming from? Where's it going to? I don't recognize it," you may want to look a little bit deeper and find out what's really happening on that particular link of communication between those two machines.

**Tags:** certification, comptia, https, hypertext transfer protocol secure, secure sockets layer, security, ssl, tls, transport layer security

**Category:** CompTIA Security+ SY0-401

## **Storage Area Networking – CompTIA Security+ SY0-401: 1.4**

We have more storage than ever in our data centers. In this video, you'll learn about **NAS, SAN, Fibre Channel, and iSCSI storage technologies**.

On the whole, we are storing more data than ever before and the numbers continue to increase. From a security perspective, this becomes extremely important because a lot of this data is being transferred across the network. When we talk about storage that's across the network, we tend to use two terms almost interchangeably, but these two terms are actually very different.

One is **Network Attached Storage, or NAS**. The **NAS storage** is storage that is outside of our device. We're connecting to across the network, but we access the data on that storage at a file level. If we need to change just part of a file, then we have to overwrite the entire file on that storage device. And likewise, if we need just a little bit of data out of a file, we have to retrieve the entire file from that device to be able to work with it.

Another common term you'll hear for this remote storage device is a **SAN, or a Storage Area Network**. It is indeed a storage device that is located across the network. But under the surface, it works very differently. A **SAN** works on something called block-level access. This is very similar to how our local hard drives and storage devices work on our local computers, where if we need to change part of a file, we simply change the individual bytes within that file that we need to change and we leave the rest of the file untouched. Works exactly the same with a SAN, except we're performing that communication across the network. And as it sounds, it's much more efficient for reading and writing, because you're only changing or you're only reading the information that you need at that particular time.

One very common thing for both of these technologies is that they use a lot of bandwidth. You're storing information across the network and every time you want to send a file or receive a file, you're going to be using a lot of bandwidth on that network. It's very common to engineer these types of networks so that they are on their own isolated network that has no effect on any of the other network traffic in your organization. And it's not unusual to see very, very high speeds dedicated to this **Storage Area Network** or the **network-attached storage**.

The need for such high rates of speed across these storage networks has really driven the creation of a specialized topology called **Fibre Channel**. This **Fibre Channel technology** connects directly from a server with a **Fibre Channel port** to the storage, which is on a, also of course, a **Fibre Channel port**. And these are very high rates of speed. You can run from two gigabits per second all the way up to the modern versions of **16 gigabits per second** over that **Fibre Channel link**.

Although the initial implementations of Fibre Channel ran over fiber optic technology, today's modern version of Fibre Channel will run over both fiber and copper cables. Just as **ethernet** has switches that support the communication across the ethernet **topology**, Fibre Channel also has Fibre Channel switches that everybody connects to. So if you have a server that needs to connect to Fibre Channel storage, then you will need a Fibre Channel port somewhere on that server.

Often very high end servers will have a Fibre Channel interface already built into the motherboard. But you could, of course, add an adapter card to provide that interface as well. Servers are often referred to as initiators, and the storage devices themselves are referred to as the targets on a Fibre Channel topology. The communication between the initiator and the target is often over very well known technologies like **SCSI, serial attached SCSI, or using SATA commands**.

On a **Fibre Channel** storage network, you would ideally connect directly to the **Fibre Channel switch**. But if you do have devices that are outside the network or still need access to the **Fibre Channel storage** but don't have a **Fibre Channel interface**, you can run **Fibre Channel over Ethernet, or FCOE**. This communicates and sends **Fibre Channel messages over an ethernet network** and it doesn't require your workstation or your server to have a specialized Fibre Channel interface. This is usually something that is integrating to an existing Fibre Channel infrastructure. So there is usually an ethernet connection coming out of your fiber channel switches that provides this link between the Fibre Channel world and the ethernet world.

**Fibre Channel over Ethernet** is a non-routable protocol that's using the ethernet frames as communication. So it's something that you commonly see within a single subnet or a single local area. You don't often run this type of technology over larger distances where all of that traffic would be routed.

Of course, there's a solution for sending Fibre Channel information over these routable IP networks, and that's called **Fibre Channel over IP, or FCIP**. **Fibre Channel over IP** is taking all the Fibre Channel information and encapsulating it within the TCP/IP packets themselves. This is sometimes referred to as Fibre Channel tunneling, because we're putting all the Fibre Channel information and tunneling it through that IP network.

This allows us to have devices that are very geographically dispersed across multiple locations and multiple data centers, but still able to send information and use the storage network on the Fibre Channel infrastructure.

Another popular technology for connecting you to your data across the network is called **iSCSI**. **iSCSI** stands for **internet small computer systems interface**. If you've ever worked with SCSI drives on a local computer, this is a way to extend that technology across the network through a routed set of protocols. It's a standard that was created by **IBM and Cisco**. And it's one that, instead of being proprietary, is very open. There's an **RFC standard for iSCSI**.

Just like Storage Area Networks and Fibre Channel, **iSCSI** allows you to use the storage across the network, but make that storage look like it is on your local computer. That block-level storage means you have very efficient reads and writes to that storage. And because it's SCSI, it's something that is very well known in the industry. SCSI's been around for a very long time. And the commands used to access SCSI devices are ones that the developers are very comfortable with. Drivers are available for iSCSI across many different operating systems, and it's quite easy to implement because you don't need any proprietary hardware or software to make iSCSI work.

**Tags:** [certification](#), [comptia](#), [fibre channel](#), [https](#), [iscsi](#), [nas](#), [san](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **NetBIOS – CompTIA Security+ SY0-401: 1.4**

The **NetBIOS** family of protocols has been used for years for many different purposes. In this video, you'll learn about various forms of **NetBIOS** and how they are used in today's technologies.

**NetBIOS** stands for **Network Basic Input/Output System** and it's a technology that has been around for a very, very long time. Technically, **NetBIOS** is an **API**, it's an **application programming interface**. It's a structure that developers can use to create applications that will use the standard way of communicating across the network.

It is something that we've seen implemented in the Windows operating system, most recently using something called **NetBEUI**, that stands for **NetBIOS Enhanced User Interface**. Sometimes, you'll see this referred to as **NetBIOS Frames or NBF**. This was a type of communication that Windows used to talk between systems but it was used prior

to Windows XP and it was not routable. If you wanted to use this technology you had to all be on the same subnet and, obviously with our very distributed and diverse networks, being able to communicate on a single subnet was very limiting. Microsoft updated this communication mechanism to use **NetBIOS over TCP/IP**. You'll see this also referred to as **NBT**. This is what you will see if you look at how Windows is communicating between devices because it's putting the **NetBIOS** information within a **TCP/IP packet**. Because of that we're able to then route outside of our subnet if we need to.

You'll see different protocols used for **NetBIOS**. You'll see the name service being used on **UDP/137 and occasionally on TCP port 137**. There's also a datagram service, which allows Windows to transfer information over a connectionless communication and, as that implies, it's using **UDP over port 138** to do that. There's also a connection-based mechanism that Windows can use to transfer information and it is a Session Service, where you are setting up a session between devices. This is then going to use **TCP over port 139** to accomplish that.

If you wanted to see some of this **NetBIOS** transferring across the network, you could bring up a Wireshark session and just start gathering information and if there is a Windows device on the network you will almost certainly be able to capture some information very shortly. In fact, I did this very thing 36 seconds into this trace file. I gathered a lot of NetBIOS communication between Windows systems and I've highlighted this single packet here which shows this browser protocol within NetBIOS that's communicating. There's a host announcement of the host Prometheus on my network. And you can see that it is a TCP/IP packet. You can see that it's running on this ethernet connection. The source IP connection is 10.1.10.12 and it's communicating out as a broadcast to the network over 10.1.10.255. This particular packet is using the user datagram protocol, or UDP, UDP port 138, so we know that this is going to be a connectionless communication and we can see that it is indeed using the NetBIOS datagram service because that is the datagram service that runs over UDP port 138. And as we look at the decode we can even see that this is a host announcing itself to the network. This particular host name is Prometheus and it happens to be in a workgroup called SGC.

There's a number of different types of communications that NetBIOS uses in Windows to transfer files, to announce itself to other machines, to be able to communicate out on work groups and domains, and it uses this NetBIOS protocol to be able to accomplish all of that, all of it running on TCP/IP.

**Tags:** [certification](#), [comptia](#), [netbeui](#), [netbios](#), [security](#), [tcp/ip](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Common Network Ports – CompTIA Security+ SY0-401: 1.4**

We rely on the **TCP and UDP protocols** to keep all of our network traffic going to the right place at the right time. In this video, you'll learn the differences between the two protocols and some of the most popular port numbers in use today.

On IP networks, there are two primary protocols that are used to get data from one side of the network to the other. One of these is called **TCP**, or **Transmission Control Protocol**. This is what we call a connection-oriented protocol.

That means that there is a very formal connection made between devices. A communication is sent, or packets are sent over that communication. And then there is an official teardown process so that everybody knows what has happened. It's very similar to picking up a phone— calling a number— you're connecting to someone, and then you hang up the phone.

This is something that also requires acknowledgments. When you send data, you're expecting to get an acknowledgement back from the other side. This is a picture of this

happening, where my laptop is sending data to a server and the server sends me back a little message that says, I'm just acknowledging that I have received the data that you've sent.

This means that the delivery of this information is very reliable. We know if the other side received that message, and if we don't get that acknowledgement back, we can simply send that data again to get it to the other side. This means that there's a lot of control that the end stations have when sending and receiving information. Sometimes when you're sending information across a network, it may end up on the other side out of order. And with **TCP communication**, those packets can be put into the correct order again so that end station understands the information that we are sending.

You can think of TCP as working with a moving truck. On one end of the communication, you're loading up the truck with a lot of different boxes and you're creating a packing list of everything that happens to be in that truck. You send the packing list in the truck to the other side.

And when you're unloading the truck, you still go down the packing list, and you make sure that all of the boxes that are coming off the truck happen to be there. If any box happens to be missing, you can call back to the other side and tell them, I'm missing one of these boxes. Box number 10 didn't make it to the other side. Could you simply send box 10.

There's no reason to resend the entire truck of boxes. I only need that single box to be sent. And TCP sends that single box to the other side. And all of the boxes are then put together in its original form.

The other common transport protocol that you'll see on our networks is **UDP**. That stands for **User Datagram Protocol**.

**UDP** is connectionless. There's no formal handshake process. There's no numbering of the traffic as it goes across the network. And there are no acknowledgements. When you send the UDP data, you have no idea whether that traffic made it to the other side or not.

We call this an unreliable protocol because of that. In fact, it's not really more reliable or less reliable than TCP happens to be. It's your information that you receive about the transmission that makes it unreliable, because we have no idea if it got to the other side or not.

There's no reordering of packets. There's no retransmissions. Once you send the UDP data, it goes through the network and you hope that it made it to the other side.

This is definitely not like a loading and unloading of a truck, because if it was, you wouldn't really care about the cargo. You're throwing it all on the truck, and you're just sending the truck off. You give it an address and you never hear whether the information in that truck ever made it to the other side.

You would think that UDP would be a very bad protocol to use on our networks. But, in fact, it's got some very specialized uses where it excels. A good example of this is voice-over IP.

Voice-over IP—very time sensitive. As I'm talking and you're listening to this traffic, there's no time to rewind and ask for a retransmission of part of the data that I've sent. When you lose a piece of voice communication, you've just lost it, and we simply continue on. UDP is a perfect protocol for using something like **voice-over IP**.

When these **TCP or UDP transport protocols** communicate across a network, they not only need to know the IP address of where they happen to be going, but they also need to know where inside of that computer they should be going. And the location inside of

that computer is something called a port number. This is different than physical ports that happened to be on an ethernet switch. These are virtual ports that are inside of a computer.

Usually a server has both an IP address and a server port number that's being used by an application. For instance, web servers often use port 80 as the number that signifies where all the web traffic will go. On a client machine, when you're sending the traffic, you're also sending it from an IP address, and you're sending it from a source port number as well. So there's actually two separate IP addresses and two separate— and most often very different port numbers— that are being used on both sides of the conversation.

With port numbers, you may often hear the term “ephemeral” and “non-ephemeral.” Non-ephemeral ports are permanent port numbers. They are not temporary. They are locked in.

Those are usually port numbers associated with a service. For example, the web service I just provided was port number 80. And that port number on that server will always be port 80. That is a non-ephemeral port number.

On your client device, you're simply using random port numbers to connect to that port 80. These are often called ephemeral ports because they are simply temporary port numbers that you're using just to establish a communication. Once that communication is over, that port number is discarded. And if you need to again communicate to that device, you pick a new port number.

This is really something that happens in real time on the client device. And it's nothing that you, as an end user, need to worry about. It's handled automatically by your network stack.

These port numbers— whether they're TCP port numbers or UDP port numbers, and whether they are ephemeral or non-ephemeral— are using a number between zero and 65,535. Most of the time, those services that you're using are using these non-ephemeral port numbers. But that doesn't necessarily have to be the case.

I chose port 80 on a web server because generally all the web servers on the internet are using port 80. But if you'd like to use a different port number, you certainly could. You would just have to make sure that all of the people using that server knew what the port number happened to be. You can imagine across the internet that could be a difficult thing to manage, which is why we use those non-ephemeral port numbers, because we always expect a web server to be running on TCP port 80.

Also keep in mind that changing the port number doesn't make things more secure, or even less secure. This is simply a way to communicate to a device and know exactly what we're communicating with. Changing port numbers around won't hide a service or provide it with any additional security.

You may hear these service port numbers, or non-ephemeral port numbers, referred to as “well-known” port numbers. There's a very large list of well-known port numbers on the internet so that your browser naturally knows to go to port 80 TCP on a server. It knows that if it needs to communicate to the service, it's always going to be waiting on that well-known port number.

Also keep in mind that TCP port numbers are different than UDP port numbers. TCP has a range of port numbers between zero and 65,535. UDP also has a range of port numbers between zero and 65,535. But a TCP port 80 is completely different than a UDP port 80. Those TCP and UDP protocols live in completely different worlds.

Here's how these port numbers might be used on the internet. On this picture I have my computer on one side, and on the other side is a server that's providing web services. But

it's also providing other services as well. And I know that because there are three well-known port numbers that happen to be configured and enabled on this device through the software running on the server.

The server happens to be running some DNS software, and that DNS software is waiting for people to make a DNS request on UDP port 53. This is also a web server, so it does have in the clear, unencrypted web traffic communicating back and forth to the server over the well-known port of TCP port 80. And this web server is also providing encrypted web services. And there is a completely different well-known TCP port number for encrypted web communication over TCP port 443.

If I want to communicate to this server over the in the clear, unencrypted web service running on TCP port 80, my computer will pick a random port number as the source port number for my 192.168.0.5. It picks a random TCP port 1331 to communicate to TCP port 80. The TCP, of course, has to be the same on both sides. But, as you can see here, the port numbers can be very different between the client and the server. And that's the communication that's set up that allows me then to communicate via web services to that web server and receive web pages back in my browser.

One of the things that you'll need to know for your Security+ exam are what some of these well-known port numbers happen to be. This tends to be something that is more rote memorization. But what you'll find is once you start working with firewalls, intrusion prevention processes, and setting up these port numbers in your applications, they almost become second nature.

Let's list out some of the TCP ports that you'll need to know for the exam. TCP port 20 and 21 are used for the file transfer protocol. Port 21 is the control protocol used with FTP. And the actual data transfer occurs over TCP port 20.

The encrypted terminal program Secure Shell, or **SSH**, runs over **TCP port 22**. This is also used for **SCP**, which stands for **Secure Copy**—a very simple copy program that uses SSH to be able to transfer data. SSH is also used for SFTP, which is a much more involved secure file transfer protocol—again over TCP port 22. This has much more capabilities than the very simple Secure Copy protocol.

**SMTP** is the way that we transfer our mail. It uses a protocol called **Simple Mail Transfer Protocol**. That runs over TCP port 25.

We saw earlier that our server was running DNS, or Domain Name Services, on that particular server, and it was using UDP port 53. But as you can see here, I've listed out TCP port 53. There is a set of protocols within DNS that provide zone transfers, which transfers a relatively large amount of traffic compared to the simple zone and name lookups that are done on the UDP side. So you will see **DNS** using both **TCP** port 53 for zone transfers and UDP port 53 to be able to do the name services lookups.

We also saw in our previous example **HTTP** being used. That's in the clear, **Hypertext Transfer Protocol** that's used for web servers, and that is commonly used on the well-known port of TCP port 80.

If you have a mobile device or you're using a mail client in your operating system, it may be retrieving mail using a protocol called **POP3**. It stands for **Post Office Protocol version three**, and it uses **TCP port 110**.

If you have **NetBIOS** running on your network, this is the **Network Basic Input/Output System**—very common to see in Windows environments. It may be using **TCP port 139** to send session information across the network.

Another mail protocol that's used by clients is **IMAP**. This stands for **Internet Message Access Protocol**, and it uses **TCP port 143**. And if you're communicating to a web server

over an encrypted link, then you're probably using **HTTPS**, which stands for **Hypertext Transfer Protocol Secure**, and it uses **TCP port 443**.

And a very common **TCP protocol** used by administrators is **RDP**. That stands for **Remote Desktop Protocol**. That's a protocol that allows you to view the contents of a desktop that's on a remote device. And that uses TCP port 3389 to communicate across the network.

The only UDP protocol that you need to memorize for your Security+— this is from your Security+ exam objectives— is UDP port 53. We talked earlier about DNS and how it uses TCP port 53 for zone transfers. If you simply need to perform a domain lookup, that's a very simple communication, and it uses UDP port 53 to accomplish that.

Hopefully that's given you an introduction of how these port numbers are used on your network. And as you start working more with firewalls, intrusion prevention devices, and many other security tools, you'll need to know exactly what port numbers need to be used to be able to secure your network.

**Tags:** [certification](#), [comptia](#), [port number](#), [security](#), [tcp](#), [udp](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Protocols and the OSI Model – CompTIA Security+ SY0-401: 1.4**

The **OSI model** is a useful description of data as it flows out of one computing device, across the network, and into another device. In this video, you'll learn how the real-world maps to the world of the OSI model.

Although the **OSI model** is a guideline, it is a model. We can take aspects of the real world and fit them into the different layers of the OSI model. So I thought it would be good to take some things that we know about and start filling in the different layers, all the way from Layer 1 up through Layer 7.

If we start at the bottom, the signaling layer, when we think about that, that physical layer is really dealing with connectivity. We're talking about cables. We're talking about interfaces. We're talking about network interface cards, and in some cases even hubs, since we are talking about having or repeating information as it's passing through. And a hub is nothing more than a multi-port repeater.

The second layer or data link layer is where we would start to see the actual frames being created and put on to the network. So at the most basic level of a frame, we're talking about **MAC addresses**, at least in the Ethernet world. You may also see the term **Extended Unique Identifier**. And this is an **IEEE trademark term**, an **EUI-48** or an **EUI-64**, which refers to the format, the 48-bit and the 64-bit format, of a physical address.

We also know that switches work at this layer. When they see a MAC address, it then looks up in its table where it should be sending that MAC address. So all the forwarding decisions made by switches happen at this Layer 2, the data link control layer of the OSI model.

The **network layer** is where we start to see network addresses. And so the **OSI model** certainly expects IP addresses to be at this layer. Of course, that's where our routers work as well, since routers make their routing decisions and their forwarding decisions based on these Layer 3 addresses, these network addresses.

And we often refer to this segment of a frame as a packet. So everything at this Layer 3 and above is inside of a packet. And we're packetizing or sending this information across the network, all of this happening at Layer 3.

Layer 4 of the **OSI model** is our transport layer. And just as the name implies, these are the protocols that are transporting information from one side of the network to the other. So this is where you would expect to see **TCP**, our **Transmission Control Protocol**, and **UDP**, our **User Datagram Protocol**. And obviously, these work very differently.

**TCP** when it sends information expects to get an acknowledgement back. **UDP** sends information and has no idea if that data got there or not. No acknowledgements are sent back as part of the UDP protocols. But both of these protocols, since they are transporting information, fit nicely at this Layer 4 of the OSI model.

Our session layer is where we might see control protocols being used to set up a session or tear down a session. These are also protocols that might be used to set up tunnels between one station and another. And those are tunneled into the next layer, Layer 6, our presentation layer. If we're doing any type of encryption, if we're going to a website and we're sending encrypted data back and forth or receiving encrypted data that we need to decrypt, all of that process takes place at this Layer 6 and gets it ready to present to us, which is really going to be at Layer 7.

Finally at Layer 7 is where we see the email. We see the decrypted information, the decrypted website or web page that we were asking for. All that happens there. So every time we send information, it starts at Layer 7, works its way all the way down to Layer 1, goes across the network. And at the other end, it performs exactly the same thing all the way back up to Layer 7 again.

If we were to look at this in a protocol decode, it would almost map exactly to what we were just looking at. For instance, this is a screenshot from a Wireshark session where I just grabbed communication to I think it was Google Mail that I was doing. And in this particular case, we can see that we start with a frame. So we're really talking about this Layer 1, Layer 2 functionality. In fact, there is the MAC addresses, the Layer 2 address of the source device, and the Layer 2 address of what in this case was a Netgear router getting it ready to send out across the internet.

At the Layer 3 internet protocol, you can see my source IP addresses here. And I was indeed talking to Google Mail. And you can see the IP address of Google Mail, so that clears up Layer 3 for us.

Obviously, information is being transported within this Google Mail communication, so we're using transmission control protocol, the Layer 4 of your OSI model. Here's your source port. There's the destination port, 443, which tells me this was web traffic that was encrypted. And then you have sequence numbers, acknowledgement numbers, and linked information.

Now, above this there's not much that you can look at, because everything at this layer now up at the secure socket layer is all encrypted data. And it's not uncommon to put Layers 5, 6, and 7 into their own little block and say everything above Layer 4 is the application. It's setting up the session. It's encrypting and decrypting data. It's presenting information to you.

But that's a very good example of how you can pull the packets right off the network, have a look at how those are presented to you, and map them back to the specific **OSI layers**. And that's exactly what happened in this case is that I had an application, which was my Google Mail, which was then encrypted via **SSL or TLS**. It was sent down and transported via TCP across my network using IP addresses to communicate across those long distances. And just to get that data to my local router, I used **MAC addresses** at that data link layer to get it the hop along the way.

And of course, at last, that router was putting it on to the wire as electrical signals. Those electrical signals went across the internet, hopped through a number of routers along the way, so indeed it probably hopped up to Layer 3 and 4 along the way. And then finally,

once it got to the other side, it was able to be recreated on Google's server back up to the application layer.

So although we call this an OSI model, there are a lot of real-world connections to the way our applications work. And this is the way we'll be able to communicate with other network professionals when they're asking, where's the problem, where are you seeing an issue, you can speak specifically to, I'm seeing an issue at Layer 3 with being able to communicate to that IP address. Or I'm seeing an issue at Layer 1, because I'm having physical layer connectivity issues. All of these work together to make sure that information can go back and forth. And as you can see, it's not too hard to map what's happening in the real world right back to our OSI model.

**Tags:** [certification](#), [comptia](#), [osi model](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Wireless Encryption – CompTIA Security+ SY0-401: 1.5**

Without encryption, our wireless networks would be relatively useless. In this video, you'll get an overview of wireless encryption with **WEP, WPA, and WPA2**.

One of the challenges with using a wireless network is that you don't have the ability to differentiate on who gets to hear the wireless network— your wireless access points are radio stationed. Anybody with the right equipment can tune into the channel and hear everything that happens to be going on. So that creates a bit of a challenge when you want to be sure that your information remains private.

That's why we've created all these different encryption algorithms, and security has such a high priority on wireless networks, it's because you can encrypt the data and make sure that just the people who need to be able to decrypt the data have that level of access. Now people with the password, then, can transmit and listen. There's methods to set up additional authentication methods over wireless networks to make sure that people inside of your network have the proper credentials to be able to do that. We're going to talk about how **WEP** can be used and how **WPA and WPA2** are used to help protect data on the wireless networks that we use today.

When our 802.11 networks first came out, we came out with an encryption methodology called **WEP**— that stands for **Wired Equivalent privacy**. Like the name implies, we wanted to have the same level of protection on our wireless networks as we were getting on our wired networks. The way that this worked is you are able to set up some encryption keys in your wireless access point— either 40-bit keys or 104-bit keys, depending on which you would like to use. You can use different key sizes on your wireless access points.

The problem was that in 2001, we found some pretty significant cryptographic vulnerabilities associated with WEP. And it was basically the fact that WEP used static keys, the keys never changed. All of the people on your network were using exactly the same encryption key and the key was static all the time. So WEP became very easy to crack into a web connection using very basic functionalities of the computer in really just a few minutes. And for that reason, it's highly recommended that nobody use WEP.

Sometimes you'll run into legacy devices and all they can do is WEP. And you might want to consider just not using those legacy devices or setting up a completely different network just for legacy devices that need to communicate via WEP. Obviously when that particular encryption vulnerability was found, we decided, well, we need something quickly to replace that. So we came up with something called Wi-Fi protected access.

And you'll see this referred to as **WPA, WPA2**, and there's even a flavor called **WPA2-Enterprise**, where we are integrating with 802.1X to be able to do authentication.

The **WPA** when it first came out, **WPA** used something called **TKIP** as the ability to encrypt the data that was going by. So this was an improvement over **WEP**, primarily because this temporal key integrity protocol allowed us to be able to change the keys in every packet. And so we rotated through those keys, which made it very difficult to decrypt that data. It was harder to hack into something like that.

Every packet got a unique encryption key, so therefore, there was constant change going on. This was really a stopgap measure, there were better ways to do the encryption—stronger encryption algorithms, and we knew we needed to move to that. But at least this very simplified and very useful **TKIP protocol** allowed us to use our older hardware to be able to still maintain an encrypted connection.

But in reality, everybody really should have moved by this point to **WPA2**. And this was the final certified version of WPA— came out in 2004. What it did was allow **AES level encryption**, that's the advanced encryption standard algorithm, which is a very powerful encryption method. And it used a protocol called **CCMP**, the **Counter Mode with Cypher Block Chaining Message Authentication Code Protocol**. Boy, that's a mouthful.

But that is the fundamental protocol that's used to encrypt data inside a WPA2 wireless network. So as you're looking at your wireless settings, you may have settings that say do you want to use WPA2 with TKIP, do you want to use WPA2 with CCMP? And if you have the ability to do that and have everybody on your network able to use WPA2 with CCMP, that's probably the one you'd like to choose.

**Tags:** [certification](#), [comptia](#), [encryption](#), [security](#), [wep](#), [wireless](#), [wpa](#), [wpa2](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **EAP, LEAP, and PEAP – CompTIA Security+ SY0-401: 1.5**

Wireless authentication is handled using one of the **EAP family** of protocols. In this video, you'll learn the differences between the **EAP, LEAP, and PEAP authentication protocols**.

Now that you've decided how to encrypt the data going over your network, we now need to think about how we authenticate people to be able to use the wireless network. And there are some standard protocols you can use to do that. There's **EAP**, there's **PEAP**, and there's **LEAP** to look at.

**EAP, or eap, or extensible authentication protocol** is a very common set of frameworks that can be used to authenticate people onto things like wireless networks. For instance, WPA2 and WPA use five different EAP types as authentication mechanisms. A very common way of setting up the authentication methods, especially early on in wireless networks, was created as a proprietary method by Cisco. And it's called **LEAP**, that stands for **light weight extensible authentication protocol**.

One of the nice things about **LEAP**, and the reason that it's called light weight, is that you don't have to set up any digital certificates whatsoever. There's no PKI involved. You simply use passwords and you're able to communicate between your authentication methods and your wireless access points. This is based on **Microsoft CHAP**, which means that the information that's being sent between these devices has a few security shortcomings.

A large amount of this traffic is in the clear. Even if it's being hashed, you're still able to see it without any special type of encryption going by. So most people think, eh, they would like a little more encryption on their wireless network, especially for their

authentication. Most of the time then, you'd be implementing something like **PEAP**, which stands for **protected extensible authentication protocol**.

This was created by Cisco and Microsoft and RSA Security to come up with a way to encrypt all of this communication. That's very much a standard and it networks across many different wireless devices. What this essentially does is create a **TLS tunnel**. Most people think of this as an **SSL tunnel**, which means you only need a certificate on the authentication server. And that way the authentication communication is all encrypted within that tunnel.

**Tags:** [certification](#), [comptia](#), [ear](#), [encryption](#), [leap](#), [peap](#), [security](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **MAC Address Filtering – CompTIA Security+ SY0-401: 1.5**

One common method of access control on a wireless network is the use of physical hardware addresses. In this video, you'll learn how to limit wireless access using **MAC addresses**.

We've now encrypted our data. We've chosen an authentication method so we can make sure that people get on to the network who are allowed and prevent those who aren't. There may be some other things that we can configure in our wireless access point to help with security. Let's look at one of these. This is **MAC filtering**. **MAC** stands for **Media Access Control**. It's the hardware address, the Mac address, of your wireless card that's in your devices. And one of the things you can do is list out in your wireless access point a list of all of the **MAC level addresses** who are allowed to communicate to your wireless network. So this keeps all your neighbors out. This keeps other people who don't your MAC addresses from being able to even communicate to your **wireless access points**. This obviously creates a lot of administration. You have to get a list of everybody's MAC address and put it into the wireless access point. If you have visitors that are coming in, you may have to add those MAC addresses also to the wireless access point.

And in reality, as long as somebody has a protocol analyzer they can sniff what's happening over a wireless network. Becomes really easy to find out what MAC addresses might be out there, and it's really easy to spoof MAC addresses. So we can simply wait until you leave for the day and then use your MAC address to get back on the network. So obviously that's not the only security feature you should apply to your access points and your wireless routers because it's so easy to do this. We call that security through obscurity. And in reality, that is not the security at all. There's no real security if all you're doing is trying to hide something that somebody later on can very, very easily find out that information. It's only going to protect you from the people who don't want to get into your network to begin with. So don't think and use MAC layer filtering as the only security method. Use it to layer on along with all of the other things that you're doing.

**Tags:** [certification](#), [comptia](#), [encryption](#), [filter](#), [mac address](#), [security](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **SSID Management – CompTIA Security+ SY0-401: 1.5**

What's the name of your wireless network? In this video, you'll learn how to limit the advertising of your wireless SSID.

If you've ever noticed when you searched for a wireless access point, you were able to find the name of the access point. It pops it up on the screen. What access point would you like to connect to? Maybe LINKSYS, or DEFAULT or NETGEAR or a name that's very specific that someone has programmed into their wireless access point.

Well, being able to identify wireless access points so easily and connect to them, it also brings up some security concerns. Should we really be broadcasting the fact that we have a wireless access point here?

So one of the things you can do is, of course, change the **SSID**, the **service-set identifier**, to something that's not quite so obvious. Make sure it doesn't use a default name like LINKSYS and maybe not even give it a name that's referring back to your organization. Give it something very generic.

You can also disable the broadcasting completely. This is a configuration setting from my access point. Here's a check box, Enable SSID broadcast or not. I can turn it on or off.

But yet again, it's very easy with protocol analyzers to be able to sniff the air and see what access points are out there. As soon as somebody connects, I'm going to see the SSID. You can't hide all of the SSID information.

So, again, applying this is really security through obscurity, which, of course, is nothing to do with security. So don't, again, use this as your only method of trying to add extra security to your network. Layer it on with all of the other things that you're doing.

**Tags:** certification, comptia, encryption, security, service set identifier, ssid, wireless

**Category:** CompTIA Security+ SY0-401

## **TKIP and CCMP – CompTIA Security+ SY0-401: 1.5**

The **TKIP and CCMP protocols** have been an important part of our wireless key management and encryption technologies. In this video, you'll learn how TKIP and CCMP relates to WPA and WPA2 wireless encryption.

In this video, we're going to talk about the technologies used for encryption on wireless networks. Specifically on networks using **WPA, WPA2, and WPA2-enterprise**. **WPA** stands for **Wi-Fi Protected Access**.

And you'll notice we're not going to talk about **WEP**, which stands for **Wired Equivalent Privacy**. The older WEP encryption that was used on wireless networks was found to have some cryptographic flaws, and therefore you should not– and certainly probably won't– see WEP used on today's modern wireless networks.

We're going to focus on two types of technologies used in WPA. The first one we'll talk about is **TKIP**. That stands for **Temporal Key Integrity Protocol**. TKIP was built to rotate keys around so that there would not be the same problems we ran into with encryption with the **WEP protocol**. And TKIP was also something that made sure that there would be something unique about each one of these encryption keys.

The other technology that we'll talk about is the one we commonly see with WPA2 today, and that is **AES** that is used in conjunction with **CCMP**. **AES** is the **Advanced Encryption Standard** algorithm that's doing a lot of the encryption. And it's combined with **CCMP**, which is **Counter Mode with Cypher Block Chaining Message Authentication Code Protocol**. We hardly call it that because it's so many words. You most often see it referred

to when you're setting up your wireless network as **WPA2** and in parentheses it might have **TKIP** or it might say **AES and CCMP**.

When we ran into the cryptographic problems with the WEP protocol, we needed something to fill the gaps. And so we created TKIP. This allowed us to make those 802.11 networks more secure without worrying about the cryptographic problems that we had with WEP. One of the keys with TKIP is that it makes the keys together. It took this secret root key and mixed it with the initialization vector. And this made the key much more secure because it was constantly changing.

Another nice edition of **TKIP** is that it includes sequence counters. This is useful to avoid replay attacks. In a replay attack, someone can record the information going over the network and then replay it again to gain access. Instead of having you there, they would pretend that they were you because they were replaying your previous content. Well, with a sequence counter, you can't replay content because it would still have the old counter numbers inside of it. So this was one way to make sure that no one could record that and then use that information later.

**TKIP** also implemented a **64-bit message integrity check**. This meant that information could not be changed somewhere in the middle of the conversation. This is a big problem if you're worried about a man-in-the-middle attack where someone would receive information, modify that information, and then send it on to you. With the message integrity check, you can be assured that the original information is still intact when it gets to you.

We see TKIP being used with the WPA encryption protocol. This was the stop gap between WEP and WPA2. With the WPA2, we chose to go a different route with encryption. That different route with encryption implemented **CCMP**, the **Counter Mode with Cypher Block Chaining Message Authentication Code Protocol**. This is what replaced TKIP when the final WPA2 implementation was released. This was a more advanced encryption standard. It had a larger key size, it had a larger block size to be able to do the encryption, and it used a lot more computing resources. It used encryption algorithms that required more CPU usage. And we usually solve out this time frame that many people had to upgrade their wireless hardware to be able to implement WPA2. These days, our hardware is up to date and we generally see WPA2 used on all of our wireless devices.

There were some nice capabilities added with CCMP. One of them was data confidentiality, where only certain people that were authorized to receive information across the network could receive that data. There's also authentication enabled within CCMP, so you can be assured that the user on the network really is the genuine user. There's also access control implemented within CCMP. So we were able to allow or disallow access to the network based on your credentials.

If you're working with some older hardware, you may see that it only supports WPA and not WPA2, and therefore would only be supporting TKIP. On newer access points and wireless devices, you may see those supporting WPA2, which of course would be supporting CCMP and AES. And you may see on the newer devices that there might be options to support some of the older hardware, so you may also be able to even configure the newer hardware to simply use WPA. But as our older hardware is phased out, these days we tend to always use WPA2, which means we're going to be using CCMP in combination with AES.

**Tags:** [ccmp](#), [certification](#), [comptia](#), [security](#), [tkip](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

**Wireless Power and Antenna Placement – CompTIA Security+ SY0-401: 1.5**

Configuring a wireless network requires a combination of power settings, antenna choice, and antenna location. In this video, you'll learn how power and antenna settings can be used to customize your wireless network installation.

A feature that can really help you with security in your wireless access point configuration, you may have controls over how much power you put out on the wireless access point. Ideally, you would set this to go as low as you possibly can and still have people communicate. That way you aren't sending your signal out to the parking lot where other people may be able to hear what's going on on your wireless network.

So you may have to study it. You may have to get some detailed spectral views of the traffic. Determine what type of traffic we're seeing, how much power is really being outputted, and determine, based on the size of the organization or the size of the floor you happen to be on, what is the level of power you really should set.

You should also think about the receiver. If somebody has a high-gain antenna in the parking lot, they may still be able to hear things, even though you've set the power down really, really low. So this is really going to have a dependency on where you're located. And you're going to want to set the power settings accordingly.

Obviously, this is not going to be the only thing that you would set to be able to limit access to the wireless network. But, again, try to keep it away from other networks. And try to keep it as quiet as possible, as low power as possible, but still allow you to operate properly.

Along those same lines, it really does make a difference where you put the antenna for your wireless access point, especially if you need to overlap different parts of the organization. You may have a big floor. And it may not be possible to put a single wireless access point in the middle and try to see if everybody can hear that access point.

Instead, you may need to layer access points and even overlap some of the channels just a little bit. Your wireless receiver in your laptop or your wireless device will choose whichever signal is the loudest. And you'll overlap it with different channels, as well, so that you don't have frequencies that are overlapping each other and creating any problems.

So this is where you may want to adjust power levels, adjust where your different antennas are being placed, and maybe even change the type of antenna you're using, maybe not to be an omnidirectional antenna. Maybe choose one that only looks in different directions to send its signal and receive its signal. There's a lot of options out there. You can check with your manufacturer of your wireless access point and see what types of antennas are available for the particular model that you have.

**Tags:** [antenna](#), [certification](#), [comptia](#), [placement](#), [power](#), [security](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Captive Portals – CompTIA Security+ SY0-401: 1.5**

One method of controlling network access is through the use of a captive portal. In this video, you'll learn how administrators use captive portals to increase the security of the network.

Although you may not have heard the term captive portal before, I suspect you've experienced one if you've ever stayed in a hotel or you've used the wireless network at a coffee shop or in a restaurant. A **captive portal provides** a way to authenticate to a network or at least have you agree to certain terms and conditions. You almost always see this on wireless networks where people are constantly moving in and out of the wireless area, and you want to be assured that the people that are signing on to the wireless network have either agreed to certain terms and conditions, or they provided a certain set of authentication to gain access to that network. This **captive portal capability** is commonly provided by a wireless management device. It may be provided by a firewall or some other device that's constantly watching the communication on the wireless network, and if it recognizes someone new, it prevents access to other parts of the network. It effectively hold you captive and provides a portal for you to be able to authenticate.

This captive portal capability usually provides you with a message inside of your browser and prompts you to agree to terms with service or provide username and password information similar to this picture that we have on this slide. It's commonly asking for a username and password. It may just be asking you to hit the OK button to agree to certain terms. There may also be other authentication factors you can implement. If you're connecting to this network from the outside, you may be required to put in a pseudo random number from a random number generator, or there may be a certificate, or other types of code you would have to implement as additional factors to provide during the authentication process. Once you provide this authentication information or you click the submit button to agree to certain terms and conditions, you're allowed access to the network. This access may be determined based on who you are. If you authenticate with an administrator and log in, you may have access to a large part of the network. If you're authenticating as a guest, you may for instance, only have access to the internet. This access may be for a certain duration of time or there may be just constant access based on how long the system sees you active on the network. Eventually, the captive portal has a timeout value that removes that access and if you want to continue on the network, you have to proceed through the captive portal process again.

**Tags:** captive portal, certification, comptia, security, wireless

**Category:** CompTIA Security+ SY0-401

### **Antenna Types – CompTIA Security+ SY0-401: 1.5**

The type of antenna can make or break a wireless network installation. In this video, you'll learn about various antenna types and when they might best be used in a wireless network.

If you're implementing a wireless network, you may need to use different kinds of antennas to accomplish this. In this video, we'll look at a number of different antenna technologies and see which one might work best for you.

A very common antenna type we see when we purchase some of our wireless access points is an **omnidirectional antenna**. This is the antenna that simply points into the air. And you get exactly the same type of coverage wherever you happen to be anywhere

around that particular access point. The **signal** is evenly distributed on all sides. That's why we call it an omnidirectional antenna.

Because no matter where you happen to be, you receive the same signal strength from that antenna. In many environments, this is a pretty good choice because you can put the antenna in a central area. And no matter where you happen to be, you're getting the same signal strength from anywhere around that access point.

The problem occurs when you want to have a little more signal in a particular area. An **omnidirectional antenna** is very good at evenly distributing that particular signal throughout. If you want a little more signal in a particular area, however, you'll need to use a different kind of antenna.

In those cases, you'll want to use a directional antenna. And as the name implies, this allows you to focus the signal into a very particular area. This also allows you to increase the distance. Because you're taking all of that signal that normally was set **omnidirectionally**, and you're now focusing it all in one particular direction.

This means that we'll be able to have a very focused signal in one particular direction for both sending and receiving. This means if there are other signals that are off to the side that are not in that direction, we're not able to send or receive to them very well. This directional antenna is really focusing all of that signal to one particular place.

The performance of an antenna is measured in decibels. This relative measurement means that if we're able to use an antenna to effectively double the gain out of a particular antenna, that means it increases by three decibels. And you can compare the decibel ratings of different antennas to get a feel for how that antenna might work for you.

One type of directional antenna is the **Yagi antenna**. This provides us with a way to really focus our signal in one particular direction. And because of that we get a very high gain from a Yagi antenna.

Another type of antenna is the **parabolic antenna**, where it can receive a lot of signals down a very wide area on this curved back of the antenna. This curved back means that as the signals hit, they're going to reflect off of that into a single point that is reflected right into the point of that antenna. So as we're receiving from many different areas, we're able to instead focus those signals down and receive much better for that antenna.

**Tags:** antenna, certification, comptia, directional, omnidirectional, parabolic, security, wireless, yagi

**Category:** CompTIA Security+ SY0-401

## **Site Surveys – CompTIA Security+ SY0-401: 1.5**

Before installing a new wireless network, you'll probably want to perform a site survey. In this video, you'll learn why a site survey might be one of the most important parts of your wireless network installation.

Implementing a wireless network isn't something that you can simply click a button and have the best possible performance. You have to understand the wireless spectrum and what's happening currently before you can implement a new wireless device into that environment. One of the things you can do is to do a site survey. You need to understand what the existing wireless landscape happens to be. You can use scanning tools on a device or some specialized spectrum analyzers that can really tell you exactly what's going on with all of the radio signals in this particular area. It's very common then to initially try to understand where all of the other access points might be. You might see these pop up in an SS ID list or you might use to specialize scanner, like this one, to be able to understand where the access points are.

You may not even be in control of every single one of those. You may be in an office environment where there's different companies on different floors, so all of the wireless networks may not necessarily be something that you are able to change. That means that you'll have to work with what's currently in place and try to get your wireless network to have the best possible performance regardless of what happens to be a round you. This means that you'll want to do as much research as possible. You'll want to walk around with your wireless scanners or spectrum analyzers and really understand what all of the different radio signals are in that area. You may also want to take into consideration other devices that might cause interference like microwave ovens. This needs to be considered as well so that you can properly place the antennas for your access points. And since the only constant is change, you'll want to periodically recheck all of the settings that you have. Perhaps show up again and do another site survey, and make sure that there wasn't a new access point or another device that might be conflicting with the frequencies that you'd like to use for your wireless network.

One of the keys for us on a wireless network is to have our own frequencies in use. If there are other access points in the area, let's make sure that they're running on completely different frequencies that are not going to overlap with the ones that we would like to use. For an example, let's look at the implementation of an 802.11b network, although the same idea applies to all of the different wireless types. And we have a floor plan here that shows us where all of the different devices are, and the different offices, the different conference rooms, and let's overlay what a layout might look like of wireless access points. So in this particular case, I've overlaid where the signals might be for these Omni directional antennas. And you'll notice what I've tried to do is put channels on to this that would not overlap or interfere with the channels that are directly next to us. And for this particular case, we're using 802.11b and we know that channel one, channel six, and channel 11 are completely separate channels that do not overlap with each other. So we've chosen to use those three and notice that we've interlace them so that no channel one is ever going to conflict with another channel one. Channel six would not conflict with another channel six, and channel 11 would not conflict with another channel 11. You have other frequencies and other channel choices depending on the wireless type that you'll be using, but you'll want to use the same type of methodology, so that when you implement your wireless network, you're not going to have any problems with interference with the signal.

**Tags:** certification, comptia, security, site survey, wireless

**Category:** CompTIA Security+ SY0-401

## **VPN Over Open Wireless Networks – CompTIA Security+ SY0-401: 1.5**

If you do any work on open wireless networks, you may want to consider using a VPN. In this video, you'll learn how VPNs are able to secure all of your network traffic over an insecure open wireless networks.

Using an open wireless access point that's at a coffee shop or a hotel or a school has some significant security concerns associated with it. As you're sending information to that access point, it's being sent in the clear. There's no wireless encryption that happens to be going on because it's an open access point.

This is probably different than the access point you have at home, where you've implemented **WPA2 encryption**. But because this is an open access point and you want to have as many people using it as possible, you don't generally see it being implemented with any wireless encryption.

Because of this, everything you send can be seen by anybody else who can receive that radio signal. This means that they could be sitting anywhere nearby and be able to see everything that's happening. They can look at your data that you're sending.

They can read your emails that you happen to be looking at. They can see what websites you happen to be visiting. Even if you're visiting some sites via **HTTPS**, you're still going to have other traffic from your desktop or your laptop that can still be seen over this wireless network.

That's why it's increasingly common that people use a **VPN**, or **virtual private network**, to protect the data that's going through these open wireless access points. That means that every single bit and byte that's leaving your computer is going to be encrypted. You don't have to worry about visiting a website and using encrypted protocols. You don't have to worry about setting up your email client in a way that everything will be encrypted. With a VPN, everything going out of your computer is encrypted, whether you configured it that way or not.

Here's how this works. You've got your laptop at the coffee shop. And you need to communicate to a corporate network or back to some other location on the internet.

The way that you do this is generally install and run a piece of software on your machine that then creates an encrypted tunnel to what we call a **VPN concentrator**. This **VPN concentrator** is specifically designed to be able to handle the encryption and decryption required. So it usually has a very beefy set of CPUs that's able to perform this very quickly. You're creating this encrypted tunnel, which means everything between your device and the VPN concentrator is going to be protected. Even if somebody was to capture that data on the wireless network, they would have no idea what to do with it because they would not be able to decrypt it.

It's the VPN concentrator that then does the decryption, turns it back into the in-the-clear traffic, and sends it through to the corporate network. That's why it's increasingly common for people to use these VPN software VPN concentrators, especially when you're in an environment where somebody might be listening in to your conversations.

**Tags:** certification, comptia, open wireless, security, vpn, wireless

**Category:** CompTIA Security+ SY0-401

## **Control Types – CompTIA Security+ SY0-401: 2.1**

If you're planning to build a structured security policy, you'll find the organization of security controls to be a valuable starting point. In this video, you'll learn about the **NIST** standards for the organization of security control types.

A good place to start the conversation about risk, is with the control types. The **National Institute of Standards and Technology** is a federal organization in the United States that comes up with standards that are used not only for the Federal Government, but also nationally and even worldwide.

They have a set of standards called the "**NIST Special Publication 800-53**." And that is a publication called the "**Recommended Security Controls for Federal Information Systems**." And although the name says "**Federal Information Systems**," there's some nice information in here that you could almost apply to anybody's organization.

If you go out to Google you search for "**NIST Special Publication 800-53**," you will see this. This is the document itself. It is quite comprehensive, and it's a very nice overview and a guide to how you can start taking different parts of your organization and the different kinds of risks that you have, and categorizing them, and then setting some standards on how you can deal with risk associated with those different parts of your organization.

Inside of this document are what they call, "**Three Classes and 18 Different Families**" that are categorized in these three classes. The first class is one called, **Technical Control Types**. So you can think of this as things like access control— how you authenticate onto the different resources that are on your network or on your computer. How do you protect your systems? How do you protect your communications? All of those technical aspects of control are related in that particular technical class.

The second class is the **Management Class**. This is a class that talks about how you manage these different aspects of risk in your environment. Things like how you do security assessments, how you provide authorization to different resources in your network or in your environment, how you do planning, how you do risk assessment. Those are extremely important things when you're dealing with security. Security isn't just configuring a firewall properly, it's also setting the proper policies and procedures to follow so that the firewall can be configured properly.

The last class is an **Operational Class**. What do you do, ongoing with operation, to maintain the security in your environment? What do you do when an incident occurs? What are the proper processes to go through? How do you handle changes in configurations inside of your network? You don't want to create security issues related to changes, you don't want people making changes without authorization. How do you protect things physically? Do you lock doors? Do you have key cards? How do you lock down a laptop computer that's very mobile?

So all of that's in more of the operational mode of things, and so all three of these classes all work together. You really can't look at just one of them, you have to take into account your technical controls, your management controls, and the operational controls as well.

Here's a chart that really does summarize these classes, families, and what they call, identifiers, for each one of these. It would not be a federal document if we didn't in some way have a list of abbreviations associated with these. For the purposes of the Security Plus Exam, you don't have to remember all the identifiers, but it's useful to know these different classes— the technical, the operational, and the management. And have an understanding of why these different families are associated with these classes.

And if you start looking at them, they make a lot of sense. Technical– yep, that’s access control. That’s configuring firewalls and making sure your access control lists are set properly. Operationally, you want to have awareness and training.

So, make sure you’re aware of what some of these different families are in these different classes. Get an understanding of what these control types are, and if you have time, read through the document. It’s really a very nice overview– quite comprehensive, in fact, of all of these different families and classes and how our federal government uses these classes to control and manage their security. You could probably take some of the things they’re doing, and use in your environment as well.

**Tags:** [certification](#), [comptia](#), [control types](#), [nist](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **False Positives and False Negatives – CompTIA Security+ SY0-401: 2.1**

As you build your security strategy, you’ll need to manage the inevitable false positives and false negatives. In this video, you’ll learn about false positives and false negatives and how to handle them in your environment.

If you’re working with antivirus software, anti-malware software, or intrusion prevention systems, you may run into cases where you might get a false positive or a false negative. Let’s look at both of these situations, and see how we can resolve these particular issues. A false positive is when you receive an alert from a security device that’s telling you that there was a problem. The issue with this, is that the security device is actually incorrect. This is a positive, but it’s a false positive– which means there wasn’t really a problem to begin with.

If you’re getting a message from an intrusion detection system or intrusion prevention system, these alerts are usually based on signatures. A piece of information has gone through the **IPS** that matches a signature, and it’s informing you that there was a match to that. And generally, we have to rely on these signatures, so you always want to be sure that you’re updating to the latest signatures so that a lot of these false positives might not occur.

These false positives can also occur with antivirus or anti-malware software. For instance, in April 2010, McAfee Virus Scan thought that the Windows system program svchost.exe was a virus. Well, that was certainly a false positive– that is an integral part of the **Windows XP Operating System**. And so, it removed that file, which meant that all **Windows XP SP3** devices could not boot. You had to correct that before you rebooted the machine, or once it was rebooted, you had to go through recovery process.

In October of 2011, Microsoft Security Essentials thought that the Chrome Browser was a piece of malware called Zbot, and it deleted the entire browser. So you would try to load the Chrome browser onto your machine, and it would simply be deleted because of the false positive associated with this inside of Microsoft Security Essentials.

If you’re trying to determine if something is really a false positive or not, you might want to get a second opinion. A good choice is the now Google-owned, [virustotal.com](#)– where you can point to a particular file on the internet or upload your own files, and see what the results might be in many different types of security software.

This is the [virustotal.com](#) website, and I’ve chosen to upload a file called **GPupdate** that I received in my email. I suspect the virus writer was trying to get me to run this, thinking it was **GPUpdate**, for the **Group Policy Update inside of Microsoft Windows**. Let’s choose to scan that file, It’s going to be uploaded, and this file has been seen before by **VirusTotal**. But I’m going to ask to re-analyze this file so that we’re able to see the process that goes through.

Behind the scenes, **VirusTotal** has a lot of different antivirus software that it's going to run against this particular file. So it gives us the name of the file, and it tells us how many different antivirus and anti-malware software is going to detect this particular file as being malicious or being benign. And you can see so far, only one out of the 27 or one out of 36 that have been checked, is showing up as malicious software. Looks like we've got two now out of 54.

So as it goes through the scan, you can see the different software like **Avast!**, and you've got **Doctor Web**, and **F-Secure**, and **Fortinet**, and **Kaspersky**. You've got a lot of different software that you can choose from, but only 2 out of that 54 recognized this particular file as being something malicious.

You can also get more details on these files. It'll even run through different types of Heuristics. For instance, **F-Secure** found that this was indeed suspicious and Symantec also categorizes this as suspicious. It doesn't have an exact match for this particular file, but it does notice that this file is doing things that it should not be doing, so it generically categorizes this. This gives you at least some idea if you receive a false positive on whether a file might be something that is malicious, or whether it's something that's not going to harm any of your computers.

The **opposite of a false positive** is a **false negative**. That means that you did not receive any alerts, no bells went off, there were no sirens, but something bad actually did get through your security systems. This got right through your defenses, and it's difficult now to go back to determine if there was a false negative or not, because there's no way to really rewind and know exactly where this might have come into your network.

This is completely silent, so if you had to reconstruct how a piece of malware got into your environment, it becomes a lot more difficult. You want to be sure to check the industry test for hits or misses. Generally, antivirus software, intrusion prevention software and hardware, goes through a number of industry tests where certain files are sent through. And then you can examine how many of them were identified, how many of them had false positives, and how many of them were missed completely they can then be categorized as a false negative.

**Tags:** certification, comptia, false negative, false positive, security

**Category:** CompTIA Security+ SY0-401

### **Reducing Risk with Security Policies – CompTIA Security+ SY0-401: 2.1**

The backbone of any security strategy is the creation and management of security policies. In this video, you'll learn why policies are important and which parts of the organization will be involved in the creation of security policies

As human beings, we tend to have a love-hate relationship with policies. I'm sorry, that's not our policy. I'm sorry, we can't do that, it's not something that we do according to the policy that we have. It's often a barrier that's put in place, but from a security perspective, these policies are things that everybody is made aware of. It's things that also allow you to do your jobs, so they become very important.

Your security role starts and ends with these policies. The better policies you have, the better security you're enabled to have on your network. If you don't have very good policies, you're not going to have very much but you can do from a security perspective to keep your organization safe. So this is not something that you create and you're done, this is something that you build and you continue to build on. It is a living document that you're constantly enhancing, improving, and changing based on the way your organization is changing.

Security policies cover a lot of different areas. They might cover physical security. What doors need locks? What happens when somebody enters the building in they're a visitor,

how do you handle that person? What happens if you show up at work and you've forgotten your badge? There should be a set of policies associated with that.

Policies are also technical policies. How you handle change control on your firewall? What happens if a machine gets a virus? What if that particular machine has confidential information on it? These are all things that must be considered. And you have to make sure there's a policy, so when that particular situation occurs, everyone knows the proper procedure to go through to handle that particular issue.

There are policies for human resources. And from a security perspective, that becomes pretty important. You want to be sure when somebody is hired into the organization, when somebody is fired or leaves the organization, you need to know exactly what to do. You never want to have somebody's credentials still remain on when they are no longer part of your organization. So there's a number of things you could do from a human resource perspective.

There's also business policies. Think about the things that you do as an organization and the way things are handled. How is information that is private to the organization handled? How to handle the release of press releases and other internal documents within your organization? You need to set policies on that as well.

If you're doing any type of encryption on your web servers, on your email servers, on your database servers— there are certainly a set of certificates that have been loaded on those machines. And being able to manage the certificates— keeping them secure, understanding who has access to those, how you build those certificates out, how you roll out trusted certificate authorities in your environment— all that falls under certificate policies.

And this is something you really have to consider. Whenever you start building out encryption and decryption creation on your servers, you'll find very quickly, the management of these certificates is quite a job. And if you don't have policies set up to allow, disallow, and manage changes of those things, it can become very, very bad over a long term. Because you're not sure exactly where your certificates are, who managed them, what are the pass phrases are associated with those certificates— it becomes a bit of a challenge.

So make sure if you're doing more and more with certificates, that you have a set of policies that you follow. And like everything else, that policy will continue to evolve as your certificates become in broader use, and people are using them more on servers and on workstations.

One of the more important policies you'll run across, as one that we've dedicated a number of videos to, is incident response. That is one of the things you always hate having to deal with as a security person. But it very often, it's one the most important things you have to consider. This very, very short period of time when an incident occurs and the time you're able to respond to it, can become very, very important for preserving data, keeping things private, maintaining up time, and ultimately maybe even having legal repercussions down the road against someone who may have caused an incident in your environment.

So that's a very broad set of policies. It is one that almost certainly you're involving your HR department, your legal department, and other parts of your organization to help build, because you have to know when an incident occurs— what do I have available to me? What options are available? Who should I contact? What can I do? And the better the policy is, the better you'll be able to respond when those incidents happen.

**Tags:** [certification](#), [comptia](#), [security](#), [security policy](#)

**Category:** [CompTIA Security+ SY0-401](#)

## Calculating Risk – CompTIA Security+ SY0-401: 2.1

The calculation of risk can help you make educated business decisions related to your security infrastructure. In this video, you'll learn how to associate a dollar value to the risks in your organization.

Seems every week there's a news story about some type of security breach. And that news story says, this security breach cost the organization \$200,000. This security breach cost the organization \$1 million. So the question is, how'd they come up with that number? Where did that really come from?

Because it's not just the damage that was done during that particular event, but also all of the money that was spent on man hours and things, that it took to resolve that issue internally. There's a lot of dollars that go after the fact that then cleanup, and getting new equipment, in solving some of the problems that occurred during that particular event.

So there needs to be way to calculate risk, and a very simple way is to start with the likelihood of the risk. So you should look at the particular risk, and understand, how often do you expect this to occur maybe in a year? We'll use an **annualized rate of occurrence, an ARO**.

So if we are as an organization and we're wondering, how often can we expect our hurricane to hit our headquarters? If you're in Montana, your annualized rate of occurrence is probably going to be very small. If you're in Florida, it might be a little bit larger. It's probably going to be a lot larger, and it's something to consider because, here, we're making a best guess. How often can we expect something like this to occur?

And maybe it's not just a hurricane— how often can we expect that we're going to have someone traveling with a laptop, and that laptop is going to be stolen, or it's going to be lost? How often should we expect something like that to occur? It's unfortunate that that occurs, but we have to think about that when we're planning for the risk.

Then we can take that particular event, and if that occurs, what is a loss from a single event occurring? If it's a laptop being stolen, maybe that laptop's worth \$1,000, or \$2,000, or \$5,000. When that laptop is, stolen what is the loss associated with that? And it's not just the monetary loss of the laptop, then you also have the time lost the person is on the road without a laptop. You have to buy a new laptop, there's the purchasing process, there's dollars associated with that. You need to consider all of that in a single loss expectancy. So that particular loss occurs, what is that?

Now in the case of something like a hurricane hitting, that loss expectancy has a very broad number associated with it. There's not a set the value. So it may be a very conservative number, or very liberal number, in dealing with how broad you're considering that single lost to be. But in this example here, let's say it's a laptop stolen. It's \$1,000 in that single loss if you happen to lose a laptop

So if you want to compute what the loss would be annually— take how often your annualized rate of occurrence might be, take what a single loss might be, and just multiply those together. So if you have seven laptops stolen in a year, that your ARO. And the single loss expectancy of a single laptop's \$1,000, then over an entire year we can expect there will be \$7,000 of risk calculated based on what we know so far.

And if there's an uptick then we're going to lose more, if we have less of those occur then we'll lose less. But this is how we're calculating risk for the year. We can start budgeting for this, we can start planning for this. Maybe we want to now get insurance based on these laptops because now we're spending a lot of money.

When people lose laptops, we need to have some type of mitigation in place that's not going to cost us \$7,000 out of pocket every time. And it might also help you— get additional

software or hardware the might help you track some of these laptops if they happen to be stolen or lost, and maybe you could recover some of them. And knowing that risk calculation number now allows you to do the business case.

Well, if it costs us \$10,000 to get software to track laptops, well, that doesn't make sense—we're only losing \$7,000 in a year. So, we can probably deal with that amount of risk, and if we happen to get more stolen in the year, we can revisit it then. Very valuable numbers to have. And if somebody shows up and says, we're thinking that this might be an issue, you may have to calculate some risk associated with that. Get that into your budget, into your business planning process, and certainly into the policies that you'll be setting.

There's another important consideration when calculating risk, and that's the risk value isn't just dollars. Whenever you're dealing with any type of risk—there's loss of information, somebody loses a laptop—there might be information on that laptop that could harm the company, should it get into somebody else's hands. And that is something you really can't calculate with dollars.

There's a quantitative value that you have with your dollars, but also a qualitative value you have to consider with risk. If there is a very risky amount of information that's on a laptop, maybe that's your justification for using encryption software, full disk encryption software, on that laptop. So even if that laptop is stolen, the worst case is that we're only losing \$1,000. And even if the quality of data on that laptop was such that would cause us more risk, we're protected against that. And that may be your business justification for that. That becomes a little bit more of a challenge—you have to sit down with people and discuss how the quality of that risk changes if that particular data, or information, or event occurs—but always something to consider, especially when you're calculating risk.

**Tags:** ale, aro, certification, comptia, qualitative, quantitative, risk, security, sle

**Category:** CompTIA Security+ SY0-401

### **Quantitative and Qualitative Risk Assessment – CompTIA Security+ SY0-401: 2.1**

Risk factors can take many forms. In this video, you'll learn how to assess both quantitative and qualitative risk factors in your environment.

There are many different ways to assess the risks that might be in your environment and the resources that are available. One common thing you can do is a **Business Impact Analysis**. You need to understand what resources you have in your environment, what services that you're making available to other people, and the things that are important your organization. And then you need to think about the threats that are out there that might have an impact on those particular resources.

You need to consider how likely some of these attacks might be. You need to understand that, would this threat be something that would be very easy to occur? Is this something we are having spam come in every day that might be phishing us? Or, are these threats very uncommon threats the might be associated with operating system vulnerabilities?

Then we need to think about if the machine was attacked and brought down. And there was a problem, and that resource was no longer available, what is the impact of the organization? Is this something that is going to create a major issue for us? If so, perhaps, we need to mitigate that with some other security devices.

Maybe if we lose a particular resource, maybe a mail server. Perhaps in your environment, losing a mail server for day isn't an enormous problem. Maybe in your environment, losing a mail server is a big deal. So you need to think about what the impact that will be, should that particular resource no longer be available.

It's sometimes very useful when you're trying to calculate risk, to put it in dollar signs— to get an absolute number from it. So we want to come up with ways to quantify what type of risk we may be taking with these. We want a dollar value that we can associate with this, and that way we're able to make some business decisions based on those risks that we have.

One of the ways to do this, is to determine what the single loss expectancy might be if a particular resource was made unavailable. If that web server goes down, if we lose our database server, if our mail server is not available and that resource is not available for people, how much money can we expect to lose from that?

And then on top of that, we need to think about what should we expect, or how often should we expect that particular resource not to be available for an entire year. And what we'll do is find the annual loss expectancy, which is how much the single loss was, multiplied by an annual rate of occurrence.

How many times during the year do expect this to happen? We'll simply multiply the number of occurrences by the amount of money we would expect to lose, and that's our dollar figure for the year. That's our annual budget that we can expect to lose. And based on that, we may decide to purchase more security devices, we may decide to change the way we're providing those services, perhaps create some redundancy, or think about other ways that we can use to help mitigate that issue.

And you also have to think about though, the historical reference here. You have to think about how often did this occur in the past. And this is very easy for things like understanding how many times we've lost the mail server over the last year, but there's things you just can't plan very well for.

If there is things like, well in this particular case, a Buffalo stampede. You're not going to go down the road of calculating an annual loss expectancy of a Buffalo stampede if you happen to be in Florida. So you run into these situations where sometimes you can't exactly put a dollar figure on these things because there's no reference. There's no way to determine if this is something that might have occurred in the past or that might even occur in the future.

To help with some of those situations, we do more of a qualitative risk assessment where it's not really dollar figures, it's really people's opinions of how badly a particular problem might be for us. So we need to think about and interview people to get their perspective of the significance. If we lost the mail server, how would that impact you and your part of the organization?

We obviously don't have dollar figures we can associate with this, but some people will do a traffic light grid or some other method to be able to view this. So here's a good example of looking at the risk factor, the impact of the organization, the annualized rate of occurrence, the cost of having controls in place, and what you might think of an overall risk. And in this case, it's a red, a yellow, and a green that's here, much like a traffic light.

So you can understand having an untrained staff might have very little impact. It might have maybe a yellow, kind of a mid-range annualized rate of occurrence. And the cost of controls for that, not very expensive, your overall risk probably in the yellow range. So you could take multiple risk factors and at least put them on a high level view, so you can get a better understanding of what the risk might be.

**Tags:** certification, comptia, qualitative, quantitative, risk, security

**Category:** CompTIA Security+ SY0-401

## **Vulnerabilities, Threat Vectors, and Probability – CompTIA Security+ SY0-401: 2.1**

The bad guys are very good at infiltrating our computer systems. In this video, you'll learn about system vulnerabilities, examples of threat vectors, and how to calculate the probability of a security risk.

A **vulnerability** is a flaw or a weakness that's going to affect security. For example, if you have a door that has a broken lock, that's certainly going to affect the security of everything inside of that room. Or it may be something in software. Or maybe a file in a Microsoft operating system library happens to have a programming vulnerability inside of it that now means that people can get access to the operating system itself. Of course, just because a vulnerability exists doesn't mean that anybody has taken advantage of that vulnerability. For example, someone has to first know that the vulnerability exists to be able to take advantage. If nobody knows that the lock on the door is broken, then nobody will know they can easily gain access to that room.

It's also very common in software to have vulnerabilities that might be sitting inside of operating system software for months or even years before anyone discovers that the vulnerability even exists. In that particular case, you might think that our operating systems are wide open. But of course, to be able to take advantage of the vulnerability, you first have to know about it. And that's why we're constantly telling you to update your operating system and make sure it's patched, because we're constantly discovering new vulnerabilities inside of that software.

The **threat vector** is the path that someone takes to be able to gain access to a device so that they can take advantage of that vulnerability. This might be your computer, it might be a mobile device, but somehow that bad guy has got to gain access to be able to take advantage of that problem. You might consider something like an email. In an email, a common threat vector might be an embedded link or an attached file, and the bad guys want you to be able to click that file so that they can then gain access to your computer.

All of these things like a web browser, wireless hotspot, or a telephone, all have threat vectors. You need to protect against fake sites or session hijacks in a browser. You need to protect against rogue access points, and you certainly need to protect against social engineering over the telephone. Some of these things have technological solutions, and others may require training of people to make sure someone doesn't take advantage of that particular threat vector.

A USB flash drive, for instance, might have an executable inside of it that automatically runs when you plug-in that USB flash drive. That's a very common threat vector. And even something like physical access. If someone's able to gain access inside of your organization, they may be able to physically change data or steal data or equipment from inside your building.

There are many other ways that people can gain access, so you want to be sure you're covering the bases against all of these particular threat vectors. Some of these are more susceptible to attack than others. It really depends on the threat vectors that apply to your organization. And you also have to consider what vulnerabilities might be there. If someone gains access to email, but you're already removing all embedded links and you're already removing all the attached files before they even get to the end user, then that threat vector is not available to the bad guys.

So what's the probability that you might be affected by some of these vulnerabilities? First you have to understand what all of the potential threats might be and all the actual threats might be. So you need to have a very good understanding of the different possible vectors and understand where the different places might be inside of your environment that someone might be able to take advantage of. Really doesn't matter what the probability is, you're really trying to determine where those might be.

Then we have to identify just how many vulnerabilities do exist in our environment. This is a very large task. We have to look at all of the different operating systems we running, we have to understand what patch level they may be up to, we need to look at what applications may be in use, what services may be running, so that we can really understand what the potential might be for somebody to take advantage of a vulnerability.

Now we can start calculating how likely it might be that we would have an attack in our environment. There's no exact formula for this. You really have to look at the number of operating systems and exactly what you might have out there that's susceptible, and then understand where the threat vectors are and how someone might gain access to this. So this is going to be very different depending on the organization. But once you start examining this, you get a better idea of just how susceptible you might be to these particular threats in your environment.

**Tags:** certification, comptia, probability, security, threat vectors, vulnerabilities

**Category:** CompTIA Security+ SY0-401

### **Risk Avoidance – CompTIA Security+ SY0-401: 2.1**

There are many ways to widen the gap between yourself and risk. In this video, you'll learn some different strategies for avoiding risk in your organization.

In any organization you're going to have risk. You have risk when you walk across the street. There's risk when you drive a car. There's risk when you go into business. And the real challenge is how you deal with that risk. There's things you can do, like risk avoidance. In organization there may be things you're doing where you just make a decision that's just too risky, we're not going to do that anymore. You've seen this a lot in universities and colleges that generally have a very open and broad access to the internet. And the problem is that people are taking advantage of that and downloading copyrighted materials, and the university is being served with legal papers. They may decide you know what, having that open access to the internet is good, but we need to start avoiding that particular risk. And let's turn off the ability to do bit torrent, or the other types of peer to peer through our internet connection. You have to make that business decision on whether that's something that you can avoid, or whether from a business perspective you can continue with that risk.

Another way to deal with risk is transfer the risk to someone else. If you're concerned about a risk of a hurricane perhaps we should get insurance. So that should a hurricane hit we would at least be covered for part of that cost. And buying insurance is a very, very common way to transfer the risk that you have to someone else. Of course you have to make sure that's risk that can be transferred not everybody's going to want to give you insurance for a hurricane. Or if they do allow to have insurance it might be very, very costly, again a business decision that you have to make.

Sometimes risk is OK. You've balanced out what is good and bad about that risk, and you've just decided you know what, we're fine with that. We're not going to limit people's access to the internet or we're not going to worry about buying that costly insurance for the hurricane, we're just going to take that on ourselves. And as long as you're aware of that you can make a business decision associated with that. Accepting the risk is an absolutely proper thing to consider. There are also things you can do to help mitigate the risk. Maybe you can allow certain things to go through your network, you can allow people access to the internet, but maybe you should be scanning things on the inbound to make sure there's no viruses inside of that. That nobody's trying to take advantage of one of our servers, take advantage of a vulnerability.

So maybe we'll buy some firewalls, some intrusion prevention systems to be able to mitigate that risk. So we're spending money, we're going through business processes because we want to be able to access the internet. But we're going to put some things in place so if bad stuff comes through that link we're going to stop it right there, and help

mitigate, decrease, the risk level that we have associated with that activity. And ultimately, there may be a way to deter the bad guys from doing things that are risky in your organization, like a big dog. A big dog is a very, very good deterrent at home for security. Maybe doesn't work so well in an organization. Maybe instead you have a lot of fences, you have technical security, you have firewalls and intrusion prevention, maybe of warning signs up on the outside, that says if you're logging into this server know that we're watching what you're doing. Sometimes just having a little bit of deterrence can go a really long way.

**Tags:** acceptance, avoidance, certification, comptia, deterrence, mitigation, risk, security, transference

**Category:** CompTIA Security+ SY0-401

### **Risks with Cloud Computing and Virtualization – CompTIA Security+ SY0-401: 2.1**

Cloud computing and virtualization are powerful new technologies, but they aren't without their own risk concerns. In this video, you'll learn about the risks associated with cloud computing and virtualization.

**Cloud computing** is all the rage isn't it? It's a technology that we've named now, and it's things that we're starting to do more of because our bandwidths are getting better, people are creating resources for us in remote locations, and we're able to blend that in with what we do as a normal part of doing business.

But there are risks associated with cloud computing, just like anything else, we have to consider those risks. One is that the data that we may be putting into the cloud may be available to more people than we want. Sometimes we're dealing with machines and services that are managed by other people, they're managed by third parties. And if you're putting data out there, there's a possibility that someone from those third parties might have access to that data. So if you're dealing with cloud computing, and your data is extremely important or extremely sensitive, you may want to consider making sure that you put limits on what people are able to see. Maybe you don't put the data in the cloud. Or maybe you encrypt it when you put it in the cloud. There's things you can do to help mitigate and allow that particular risk in your environment.

Another challenge you have from a security perspective is that the actual security access to this data, or this information, is managed by a third party. If you look at something like Google Mail or Yahoo Mail, you really don't manage the security for that. You trust that Yahoo or Google is going to be able to make sure that you're mail is secure, that nobody else gets information that you have inside of your inbox. So that's a bit of a challenge, because now we're putting that trust in a third party. And if you're putting information into the cloud that's being managed by a third party, that's certainly something you should consider.

Another piece that's important with cloud computing is that these servers are somewhere else. You may just be buying a service that happens to be on somebody else's equipment. And in that particular case, you may not have a lot of control should a problem occur with that server. If the server goes down, it loses power, a hard drive fails, or perhaps you get locked out of your accounts, you don't really have direct access to be able to resolve that particular issue. Just because it's in the cloud doesn't mean it's always available. These are humans that are managing technical systems, and sometimes what happens out there in the cloud creates downtime and outages for you. You also have to keep that in mind because there is a risk from your organization not having access to your systems. If that occurs, you need to have an understanding of what that means for the organization.

Another technology that has really come on strong is virtualization— this idea of having one big monster computer. And inside of that device you can build virtual systems. Before, we used to have 20 different servers. Now we've got one big server and virtually there's

20 little servers sitting inside of it. What's nice about that is we have a lot of control over what we can do with that system. We can allocate more memory. We can give it some more disk space. We're not limited by physical constraints anymore. So there's a lot of good business value associated with virtualization.

But from a security perspective, there is an emerging set of threats coming by somebody taking advantage of that virtualization layer. That's the layer that sits on top of all these virtual systems. And the bad guys know that if they can get access to that virtualization layer, there's a potential then for gaining access to every single virtual system that might be on that physical computer. That's a pretty big concern. You might have some very important information. You might have 100 different virtual systems on a physical device. And by gaining access to that virtualization, maybe putting every single one of those systems at risk. And it's something you have to keep track of as a security professional, because those are challenges with virtualization you simply can't ignore.

There is very little control over what happens between virtual systems. They're all inside of one big computer. It's kind of hard to take a firewall and cram it inside of this physical computer and make all the different systems communicate back and forth through that firewall. There's not a lot of virtual firewall support out there in the world, and the virtual firewall support that exists today is very, very limited on what it's able to do relative to a physical firewall. So something also to consider there. You may be doing a lot more software-based firewalls, and they might be on the servers themselves. But certainly something to consider when you're moving into a virtual environment.

There are also challenges when you start looking at multiple systems being crammed into one physical device. In a data center, if it was a physical server, you had a lot of control over who accessed that server physically. You were also even able to separate these servers off into completely different areas of the data center, and some cases, into separate data centers. And that provided you with some advantages from being able to separate that out in the environment you had, both from a data perspective and physically.

When you stick everything on one system, that separation becomes a little bit harder to manage. And yes, you can manage the separation there, there are things in place that allow you to do that, but you have to make sure they're implemented properly, that different systems are moved on to different VLAN's, that physically they can't access each other. And those things are in place. It's not as easy as looking in a room and knowing everything in this room is separated from everything in the other room. Now you have to make sure in that virtualization layer that things are being managed as separate entities, and those two systems are not able to communicate with each other.

From a business management perspective, we also have to be clear about separation of duties. When everything is on one big computer, maybe all of your databases are on separate virtual machines inside this one system, separation of duties becomes a little bit more difficult. How do you separate somebody from managing one big server that happens to contain many, many, many different servers within it?

So that's something that just has to be part of your policies. If you're managing a virtual server, maybe you have multiple people that can manage that virtual server. Maybe the administration of that server is split off into other pieces. Maybe there is an overlay on top of every single one of those individual virtual machines for management and security. Something that you may have to consider implementing into the security policies in your organization.

**Tags:** certification, cloud computing, comptia, risk, security, virtualization

**Category:** CompTIA Security+ SY0-401

## **Recovery Time Objectives – CompTIA Security+ SY0-401: 2.1**

There are many considerations when working on the recovery of business resources. In this video, you'll learn about these important recovery thresholds and how to calculate the amount of uptime and availability.

Unfortunately, problems occur. This might be a hardware failure, you might have data corruption inside of some software, or there might be an attack. And you will be expected to recover from these particular issues. One of the things you may be asked to determine is how long is it going to take to restore us back to where we were? This would be the **mean time to restore, or MTTR**. You might also hear this referred to as the mean time to repair. Sometimes that number's very easy to determine. Sometimes there's a lot more variables involved, so it requires that you make a larger estimation of what your MTTR might be.

There's also a mean time to failure. Usually when things are running OK, you might want to consider how long is it going to be before something fails? It's very common to do this with things like network infrastructure equipment. We know that hardware will eventually fail. So we want to determine how long can we expect this particular piece of hardware to run without a problem. And of course, we might have some secondary pieces of equipment or run pieces of equipment in tandem, so that if one fails, we're able to take over and recover very quickly. That number between failures is the mean time to failure. That's how long we can estimate that a particular device is going to run before it has a problem.

You don't just have one device in your environment of course, you have many different kinds of devices. And so you want to get an idea of how long is it going to be before different failures occur. This would be our **mean time between failures, or MTBF**. This is obviously a prediction, but you can of course consider all of the different equipment you might have. How long it's expected to run with its mean time to failure, and then give some particular idea of how long you can expect there to be between individual failures.

Another consideration is the **recovery time objectives, or the RTO**. You have to make a decision on how you're able to recover to a certain service level. You may be able to recover and get people back up and running, but their data may not be available. Or you may decide to get all the way back up and running, and have the data available, but the time frame for doing both of those may be very different. So you have to calculate and determine how long it's going to take to get back to every single recovery service level using that RTO.

When doing your business continuity planning, you generally have to take into account the **RPO, or the recovery point objectives**. For example, you may not be able to recover every bit of data once you have an outage. Maybe you're only doing backups every day. So when everything comes back online, you have everything up to the last days of data. But maybe you were doing backups every five minutes, and you would only lose the last five minutes of data. And of course, these have costs associated with them, so business decisions have to be made on what an acceptable RPO might be.

When we start calculating availability, it's usually based on the uptime of an application or uptime of an infrastructure. And it's almost always referred to as a percentage of uptime. For example, 99.999% availability. You sometimes hear that referred to as five nines of availability.

But just how much availability is the right amount? That number is going to depend on your particular circumstances. Maybe your organization can handle more uptime or less uptime depending on what your services might be. If you're a hospital, you want to have a large amount of up time. If you're manufacturing that is not working during the evening

hours, you can have more down time because you can do more maintenance during those off hours.

It can sometimes be a negotiated value because it ties into a bonus that you might get in your particular role. If you want to do the calculation of what a percentage uptime might be based on what the actual time is, it's pretty interesting. If you have availability of 99.9999%, that means that your actual down time— usually this is over an entire year— is 32 seconds. That's a pretty aggressive availability percentage. For five nines, you can only be down for five minutes and 15 seconds during the entire year. And an availability of 99% means that you were down for a maximum of 87 hours and 36 minutes over that entire time frame.

So you can see the difference between a 99% availability and a five nines of availability is a very, very large amount. And it also means that you have to have a lot of redundancy and a lot of planning that you've put in place to be able to say that you have five nines of availability.

**Tags:** [availability](#), [certification](#), [comptia](#), [mtbf](#), [mttf](#), [mttr](#), [recovery time](#), [rpo](#), [rto](#), [security](#), [uptime](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **On-boarding and Off-boarding Business Partners – CompTIA Security+ SY0-401: 2.2**

As you start and end projects with business partners, the security process becomes an important consideration. In this video, you'll learn about some best-practices around on-boarding and off-boarding business partners.

**On-boarding** is the process of bringing a new partner into your organization. This is a little bit different than hiring a person who will be an employee of the company. Instead, this is a third party organization or a third party individual who will now have access to assets, data, and other things that are within your organization. Obviously there will be a number of legal agreements that have to be resolved before any on-boarding can occur. You're bringing someone into the organization who is not an employee so they don't have any responsibility that a normal employee might have.

Usually agreements are made regarding the type of data someone might have access to, or what should happen if there are any problems with that person on the inside of your organization. Once we've completed the legal requirements, we can get to work with putting together the technical pieces that will allow this on-boarding process to occur.

One of the first things that usually occurs is you need some type of connection to the third party. They're probably going to gain access to information within your organization, and you might need to access data and resources that are in their organization. Usually you accomplish this by building an encrypted tunnel between the locations. You can use your existing public internet connection, but instead encrypt the data on both sides so that even though the information is going across the internet, all of the data is still protected.

If all of the information and resources will be in the same data center, there's still usually a physical segmentation between your equipment and their equipment. And at the very least, there's a logical segmentation that will keep your data protected from the third party, usually with a firewall or some other type of security device in between.

Once we've built the road that will allow all this data to traverse, we need to create some way to authenticate access into either our network or the remote side. In those particular

cases, we need to put together some standard authentication mechanism. Usually this is done through a third party authentication server that's running Radius or Tacx Plus. This will allow us to create usernames that will allow people from the on-boarding organization to gain access to the internal resources of our company.

You should also audit your security controls to make sure they're working properly. The third party that's now on-boarded should have access to the data they need, but you should make sure that you're limiting that access inside of your organization. These projects that require the on-boarding of another organization may be very short term or they might last for years, but eventually the project or requirement will come to an end, and you'll need to begin the off-boarding process.

Ideally the time frame and details of off-boarding this third party should've been figured out well during the beginning phase of the on-boarding. This way you'll know exactly the process we'll go through, you'll know the frames that everyone will be expecting, and you'll be assured that the entire process would have been covered from end to end without missing any steps in between.

One of the questions you'll need to have answered is, how do you separate the systems that you've been using and return them to their proper owners? If there's a clear delineation between your equipment and the third party, then that's probably something that's very easy to determine. But occasionally on these projects, you have equipment that's being shared by both organizations. And so the off-boarding process will need to determine how to properly remove that equipment and return it to the rightful owner.

Perhaps even more important is what happens to the data itself. Who owns the data? And when you are now off-boarding, who gets to keep that data? Do both sides of the organization maintain ownership of the data? Is only one side going to now have access to that information? All of this obviously needs to be determined before the off-boarding process even occurs.

And of course, everybody needs to be completely aware of when the final connection will be terminated between the two organizations, especially if you're dealing with lease lines where a termination of a link may require 30 days or more to have that re-enabled. You don't want to turn off a connection and then realize there is more work to be done. So this is an extremely important date to consider during the off-boarding process.

**Tags:** [certification](#), [comptia](#), [off-boarding](#), [on-boarding](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Security Implications of Social Media – CompTIA Security+ SY0-401: 2.2**

The use of social media is an important part of marketing an organization, but it can bring a number of security concerns. In this video, you'll learn about security concerns in social media and you'll discover how the lack of security on a social media account caused a problem for a major organization.

**Social media** is a fantastic way for your organization to communicate with partners and customers and other third parties, but of course this can create challenges, especially if there's a third party involved to help you manage that process. And in many organizations you may not even be managing your own Twitter feed or your Facebook presence at all. You may be handing off your entire social media presence to a third party who then has access not only to things that are inside of your organization, but also to what people hear about your organization.

If your organization maintains a social media account, you'll notice there is a lot of information contained in that account, not necessarily about your organization, but about the people that are following your organization. You may have now a large database of user names, of actual names of people, of their pictures and other information as well.

This is an extremely valuable amount of information that third parties would love to get their hands on. So it's very important that if you're managing one of these third-party accounts— especially in social media— that you're able to maintain the security of that information.

If you do have a third-party company that is representing you in a social media environment, then you want to be very particular about the tone that they use. You obviously want to have them appearing as your organization, but you also want to be careful that they're saying the right things and that they aren't embarrassing or saying things that are inappropriate about your organization. The physical control of your social media accounts can also be a concern. Obviously with a third party that's handling this for you, it's usually not just one person, but a large group of people who may be responsible for managing your social media accounts. And when you have that many people who have access to your accounts, sometimes problems can occur.

One of these situations occurred with the American Red Cross in 2011. This is obviously an organization that has a stellar reputation, but there was more than one person that had access to post to the social media accounts of the American Red Cross. And they sent out a message that said that Ryan found two more four-bottle packs of Dogfish Head's Midas Touch beer. When we drink, we do it right. The choice of beer aside, obviously the Red Cross was very concerned that this particular tweet had been sent out, and they deleted it. And they apologized and said, "We've made sure that, indeed, behind the scenes, we've gained control of our own account, and we're all working sober."

The folks at Dogfish Head actually took it to the next level and began working a blood drive in conjunction with the American Red Cross, so that we could use that particular mistake that was made to actually get something positive afterwards. This isn't always the case. Sometimes when these messages go out they can have very, very bad effects on how you are viewed as an organization. So it's very important that you're able to ensure the control and security of your social media accounts.

**Tags:** certification, comptia, security, social media

**Category:** CompTIA Security+ SY0-401

## **Interoperability Agreements – CompTIA Security+ SY0-401: 2.2**

Before entering into a business arrangement, it's always good to have a set of agreements that include security considerations. In this video, you'll learn about interoperability agreements such as service level agreements, business partner agreements, and more.

When working with third parties or outsource services, there are certainly technical agreements in place, but there are also legal agreements in place as well, in the form of interoperability agreements. You can think of this if you're working with a third party that provides you with web hosting, or with payroll, or with management of your firewall. In each of those cases, there's probably data that is being seen, viewed, or stored by a third party. And in that particular case there needs to be agreement of what happens with that data.

You might also be concerned with the hiring process from the third party, especially if they have access to sensitive information. You might also want to know more about the access controls that are in place so that if somebody is at the third party, they're only going to have access to exactly what's required to perform that third-party service. If you're planning to start this relationship with a third-party provider, you may want to consider getting your own legal department involved from the very beginning. That way, if there

any problems during this process, those particular issues can be resolved at the very start of this relationship and not once things have already begun.

One common agreement type is a **memorandum of understanding**. This is not a legal signed contract, but it is a memo that is sent from one side to the other that talks about things that are important to consider during this relationship— things like the confidentiality of data, or anything else that needs to be brought up and understood by both sides. If a third party is providing you with services, there's probably also going to be a service level agreement. This is going to define what the minimum is going to be for those services that are being provided to you. And it might also provide you with information about how much uptime is expected, or what the response times might be for certain issues. There's also usually penalties involved with the service level agreement, so that if the service levels aren't met, you will then be compensated in some way.

If your manufacturing product, or you're reselling someone else's manufactured product, you might also have an agreement in place called a business partners agreement. This particular agreement defines the role of each side and defines during this business process what the terms are for reselling the equipment and the restrictions that might be placed on a reseller. If you're part of the United States Federal Government and you need to connect to a third party, you might also be required to have an **interconnection security agreement, or an ISA**. This ensures that the connection that's being built between your part of the government and the third party will have the proper security controls in place to make sure that all of that information will stay secure.

**Tags:** [agreements](#), [bpa](#), [certification](#), [comptia](#), [interoperability](#), [isa](#), [security](#), [sla](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Privacy Considerations and Data Ownership with Third-Parties – CompTIA Security+ SY0-401:**

### **2.2**

When third-parties share data, there should always be a consideration towards privacy and data ownership. In this video, you'll learn about protecting customer privacy and the disposition of stored data.

An individual's privacy is an obvious concern, especially in the world of social media and the internet. These concerns are not only of personal privacy, but also privacy when you're working as a professional. One interesting aspect of personal privacy are the laws that have been created in Europe, where some laws not only protect you at home, but also prevent your employer from tracking what you do while you're at work.

In your organization you probably have a lot of information that you've gathered from your customers. This information also has privacy concerns, even information that seems very unimportant can be combined with other types of data to create privacy concerns. It's also important when you're working with a third party, that agreements are in place about the data and especially how the privacy of the data is to be handled.

It's often said the data is the single most important asset in an organization. If you were to lose all of your data, you could simply close the doors. The company would no longer be in business. But with all of that data in place, it's important to consider the privacy of that data. You first have to understand how you're going to protect that data and keep it private. And then you have to consider what technical requirements are going to be associated with that protection.

There should also be a consideration of the physical controls— to keep people out of the physical room where those databases may be located. And from a third party perspective, it's important to know what happens to the data once the business relationship is over. Is that information simply removed from the database? Does one particular part of this ownership have access and own the data itself, or do both sides continue maintaining

and owning that data? These agreements should be in place from the very beginning, especially when working with a third party.

This meant that all of these searches had numbers associated with the users so that they could be tracked and the research could be done. This data was of course ideal to be able to understand about the types of searches that people were doing. But unfortunately, a number of searches themselves contained PII, or personally identifiable information. Although the name was a number, the information within the search allowed third parties to then determine who exactly was making that search. And very quickly, the New York Times was able to identify very specific users, and even contacted them.

For instance, they found that user **4417749** was **Thelma Arnold, who was a 72-year-old widow who lived in Lilburn, Georgia**. And they were able to see all of the searches that that user had done. And they did an interview with Thelma, and learned what she thought about her privacy and how this information was provided out to the internet.

Of course, upon seeing these types of correlations between the searches and the actual users, **AOL** pulled that information. But of course by then it was too late. And ultimately, the chief technology officer, the researcher, and the researcher's supervisor were fired from America Online. This particular example, of course, underlines the importance associated with privacy, especially when data is provided to a third party.

**Tags:** certification, comptia, data ownership, privacy, security

**Category:** CompTIA Security+ SY0-401

### **Risk Awareness with Third-Parties – CompTIA Security+ SY0-401: 2.2**

How can an organization manage risk when a third-party is involved? In this video, you'll learn the fundamentals for risk awareness when working with other organizations.

When working with third parties you should always have some aspect of risk awareness. You're obviously connecting two systems together when you're working with a third party. And hopefully, the technical part of that has gone off without a hitch. But of course, you have to consider the security aspects of that as well. Ideally everyone gets together before this connection is made, and you all agree on exactly what security controls are going to be in place once this connection is alternately made. Both sides also have to understand what the risks are, because ideally one side or the other is going to be opening themselves up for some type of security or privacy concern.

Both sides of the business relationship probably have certain security policies in place, and obviously those security policies must be followed. But when you're working with a third party you have to balance together what resources you'll have available. You have to understand what the business requirements of this relationship are, and then you have to understand the risk of all of those things and balance them all together. Very often this risk is managed through the use of agreements. And when these agreements are in place everyone should have an understanding of the risk, and how these risks are handled throughout the relationship.

A good example of this might be something dealing with data backups. If you have a third party who's providing data into a database and you happen to own the equipment that is holding the database, who is now responsible for backing up that data? Once the backups are made what happens to that information? Is it stored on site? Is it stored off site? Is a copy sent to the third party? And then who has access to that backup data? And when you store the backup data now where is that information going to be stored? Who will have access to that storage facility? And how will that data be retrieved if you ever need to get information off of the backups?

That's just a single aspect of how the data is managed between these third parties and it's only dealing with backups. You obviously have to consider the risk for the entire

business process not just the backups. And that's why it's important to have all of these risks determined from the very beginning. Especially, when you're working with a third party.

**Tags:** certification, comptia, risk awareness, security, third parties

**Category:** CompTIA Security+ SY0-401

### **Data Ownership and Unauthorized Data Sharing – CompTIA Security+ SY0-401: 2.2**

If you've partnered with another organization, there will probably be data that is shared between you and the third-party. In this video, you'll learn how to manage data between parties and what can happen when shared data is not properly secured.

We've talked a lot in this video series about the ownership of data. That's because who owns the data is an extremely important aspect, especially when working with a third-party. When there's more than one person involved, you have to know who the ownership of the data happens to be. Is part of the ownership owned by one person, and part of the ownership of the data by another person? This is one of the problems that has to be resolved prior to putting the partnership in place. That's why in the very beginning of creating the business relationship, there needs to be a clear line of delineation and understanding about who owns the data. You also have to understand what happens to the data once the business relationship is over. And ultimately, if the data does need to be destroyed, what is the process in place and who handles the destroying of that data. When you're in a third party relationship. There is certainly going to be data that's shared between the organizations. There's probably going to be network connections in place so that this data can be shared very easily. And you do have to make sure the proper controls are in place. If you're accessing data at a third-party, you should only be able to access the data that's important for your particular business function. You should not be able to access other types of data or even other systems at the third-party organization. So it's very important that you are able to audit and ensure that these data controls are in place.

If this data is not being shared with a partner, but is instead being shared with someone else who might be outside of the organization, there's usually an agreement in place with the owners of the data so that if you're providing this data to someone, what's going to happen to that data later on. Sometimes data is shared without the explicit permission of the end user. And in those cases, it's usually a terms of service or privacy policy that makes the determination of how that data is to be shared.

Sometimes data is shared with others accidentally. This happened when Facebook announced in 2013 that for the past year, information was made available for over six million users that was beyond the scope of what those users wanted to share. There was email and telephone numbers and other information that were at risk. This accidental data sharing occurred because of a feature in Facebook that allowed you to download your friends list and have that list local on your computer. What many people didn't realize is that behind the scenes, Facebook was going to third-party databases and getting email addresses and phone numbers that were also associated with you, even though you didn't explicitly provide that information to Facebook. And when your friends downloaded the friends list, that information was also downloaded along with that friends list.

It's these types of security controls and privacy concerns that should be thought about and considered, especially when sharing data with third parties.

**Tags:** certification, comptia, data, data ownership, data sharing, security, third parties

**Category:** CompTIA Security+ SY0-401

## **Data Backups with Third-Parties – CompTIA Security+ SY0-401: 2.2**

All of your data may be well protected in your data center, but what happens to the data on your offsite backups? In this video, you'll learn about managing data backups with third-parties and what problems you could face if your data backups are not properly handled.

Our data backups are one of those things that we don't really think about unless we need them. When we delete a file or we need to reconstruct a system that's crashed, we always go back to our backups. But the rest of the time, we're not really taking them into account—at least not from a security perspective. And we absolutely should, because our backups have every single bit of data that was on our system, and now it's outside the scope of where we normally think of having that data stored.

Even if we're performing these backups ourselves, it's very often that we store this information off-site, which makes perfect sense from a security perspective. That way, if anything happens to our local environment—the building burns down or there's a flood that comes through—our data was somewhere else, and we'd be able to recover from that information. But usually this off-site facility is managed by a third party. So there's another example of how a third party might have access to this very important data.

As these data files are stored and then moved from one site to another, there are sometimes concerns about the data getting lost. There's often many, many tapes that are being transferred back and forth. The third party organization that's handling storage of that at their facility probably also handles storage for many other organizations as well. And all too often, information can be stored or filed in the inappropriate places, which makes it now difficult to retrieve later on.

Of course, the data stored on the backups may be very different depending on what you're backing up. If you're backing up a public data server so that you can restore that information later, that information is relatively open and anybody has access to it. But a database backup that contains financial information or health care records not only has very important data on it, but it should probably be handled a bit differently than information that might be public.

Here's a good example of why it's so important to manage your backup data. In September of 2011, a third-party government contractor was doing their normal backups of information, and unfortunately have the backup tapes stolen from their car. Unfortunately, this backup data was from members of the US military, and it contained health care information. This contained health care information for over 4.9 million people in the US military. And it had important information such as social security numbers, the names of these individuals, and clinical notes associated with their health care.

If you're thinking to yourself that health care information should have probably been handled differently due to federal regulations, well you would be correct. But in this particular case, the contractor believed that the oversight of this data fell under the **Federal Trade Commission** and not information that would require that they handle this information according to **HIPAA** rules and regulations.

This particular incident has resulted in at least two separate class action lawsuits—each one of them asking for \$4.9 billion in damages. This is just a very large example of things that can happen, but it really speaks well to how important it is to manage your backup data.

**Tags:** backup, certification, comptia, data, data backups, security, third parties

**Category:** CompTIA Security+ SY0-401

## **Security Policy Considerations with Third-Parties – CompTIA Security+ SY0-401: 2.2**

The process of planning and implementing security policies can provide some significant security advantages in the future. In this video, you'll learn about third-party security policies and what can happen when third-party security policies are not properly followed.

One way to protect data between third-parties is to have well-defined security policies. If you simply leave it up to an individual to do what they think is right, you may be missing some very important aspects of how to keep your data private and secure. That's because you need to protect this information between the third parties, your partners, your vendors, and even your customers. This information needs to be protected so that people aren't able to modify the information. You want to be sure the data does not get out and become disclosed to others. You want to be sure that the data is not erased or destroyed. And all of these are going to surround the plans you have in place from the very beginning regarding your security policies.

Although the implementation of the security policies is often done with technology you have to go all the way back to the beginning where there can be some contractual obligations between both parties. That way everyone knows what's expected, and what they should be doing to protect the data.

We can also think of these security policies as a living document. It's something that is constantly needing to be updated because the data is constantly changing, and our business requirements are constantly changing. It's very often that the security policies that you implement at the beginning of a project are constantly changing throughout the life of that entire project.

Although we don't know a lot of the specifics associated with the security policies that were in place during Target's data breach in November of 2013, we can still look back at what we do know and see how important it was to have security policies. What we do know about this credit card breach is that malware was installed onto point-of-sale terminals that were located within the Target stores. And when people scanned their credit cards, that credit card information was then provided back to the bad guys.

Let's see if we can backtrack over how this malware was distributed to help understand how security policies might have helped us. This Target breach was believed to have originated with a vendor of Target this vendor was infected with a PDF attachment that was sent through email. And a security policy was either in place or was not followed at this particular vendor to have anti-virus and anti-malware software running on their workstations. And ultimately, the Target vendors workstations were first infected.

The Target Corporation had a vendor network that they installed so that vendors could remotely connect into Target and provide billing information back to the Target Corporation. The bad guys took advantage of this connection and jumped from the vendor network and found the connection into the Target Corporate network. Obviously, if there were security policies in place that prevented that type of connection between the vendor network and the corporate network, there may have been an opportunity to prevent this particular breach from occurring.

There was also no segmentation between the corporate network and the networks that were at the stores. So once the bad guys gained access to the corporate network, they were then very easily able to hop to the point-of-sale terminals that were at the stores themselves. With this type of access to the point-of-sale terminals, the bad guys were able to deploy their software and then wait to collect over 40 million credit card numbers from the Target network.

It's this initial creation of security policies and the appropriate implementation of security policies that can help protect data as it's shared between third-parties. If you'd like to read more about this Target data breach, you might want to reference the [krebsonsecurity.com](http://krebsonsecurity.com)

website and learn all that we found about not only the breach itself, but reconstructed how that breach ultimately occurred.

**Tags:** [certification](#), [comptia](#), [security](#), [security policy](#), [third parties](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Third-Party Security Compliance – CompTIA Security+ SY0-401: 2.2**

The compliance of security policies is an important consideration when working with a third-party. In this video, you'll learn which security policies may be required and how to resolve issues when working with third-part security.

When you're working with a third-party, there's an additional need to comply with very particular security controls. When you have many different people accessing the same data, you want to be sure that that particular data is safe and secure. Within your own organization, security compliance has its own challenges associated with it. These challenges are even wider when you're working with a third-party. And then when you introduce new technologies like cloud computing, where your data can exist far outside the scope of both of your organizations, there are additional technical challenges you have to consider.

Sometimes this compliance is not just a good idea, it's a legal mandate. You are required by law to provide a certain level of security of this data. An example of some of these are **HIPAA**— this is the **Health Insurance Portability and Accountability Act**. You also have credit card security such as **PCI DSS**, which is the **Payment Card Industry Data Security Standard**. And for federal information security, you have the **Federal Information Security Management Act, or FISMA**.

The first step to complying with these security requirements is to understand where all of the gaps currently exist in your security. Without understanding those gaps, you're going to have no idea how to apply security controls. Now that you have your list, you can start resolving some of those security gaps.

Sometimes you can't apply a type of technology to resolve a particular issue, or resolving that problem may involve a lot of money. And in those cases you have to balance out what the business requirement happens to be with the costs associated with resolving that security concern. This security compliance needs to be checked constantly, so you need to perform periodic audits to make sure that those gaps continue to be covered and that no new problems have occurred with the security compliance. These audits can be remarkably involved and may take a long amount of time to complete. And if you're working with a third-party, you want to be sure to coordinate your efforts so that your audit goes as smoothly as possible, and you can be assured that all of your security risks have been covered.

**Tags:** [certification](#), [comptia](#), [FISMA](#), [HIPPA](#), [PCI](#) [DSS](#), [security](#), [security compliance](#), [third parties](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Change Management – CompTIA Security+ SY0-401: 2.3**

How do you make a change? In this video, you'll learn about the challenges and importance of a good change management strategy.

They say the only constant is change. That ironic phrase certainly applies to security, because there are things always changing with security. You're having to make changes to firewall policies. You need to make changes to what you're doing with switches. You're adding new systems to the network. You're removing older machines from the network. New computers are coming into your environment have to be plugged in. Older systems are being phased out. Software has to be upgraded. These things are happening all the time. And if you simply just start changing things without any type of planning, whatsoever, you could absolutely run into security issues relating to that. This is probably one of the most common risks in the enterprise, because it's something that's happening all the time. In large organizations, even medium sized organizations, this is something that occurs every week. And the organizations that have set policies in place have a meeting every week. That's your change control meeting. Everybody brings up the change in that meeting. Everybody knows what's going to happen. And now you have a window between one o'clock in the morning and four o'clock in the morning on a Sunday morning where the change takes place. And now you can on Monday see what happens when that change is now in. If you don't get your change mentioned in your change control meeting you don't get it done that week, you have to wait to the following week. It's a very common way to handle that. If you overlook this change management process you're really opening the door for some serious, serious problems down the road. Because anybody can go in make a change to any system, it's not being tracked, and eventually when a problem occurs it's a problem that has occurred because you weren't planning for that issue. You don't have a rollback in place. You don't have a way to manage that. And that's what you have to have, are clear policies associated with change.

How often are changes made? What is the duration that you're allowed to make changes? Is only going to be on that Sunday morning? What is the process to install that change in your environment? And perhaps more importantly, if it doesn't work or causes a problem what is the process to roll out of that? How long is it going to take the roll back that particular change that you've made? Often the people that need the change are not the people implementing the change. So very often there are very documented processes you have to go through to make this happen.

In the end you want to have a way to allow changes in your environment, but not restrict your business from being able to perform its duties and its functions. It's a balancing act you have to consider when you're working in a type of security environment and setting up these policies. This can be in many times very, very difficult to implement. If it's an organization that's never had any type of policy, and suddenly you show up and say, we need to follow these very specific policies, rules, and procedures there maybe a bit of push back there. Why do we have to do that? We've never had a problem in the past. We have our own way of handling these things. Let's not start creating more problems.

Corporate culture can be a very, very difficult thing to change. And of course, that all goes back to your policies and procedures that everybody gets to participate in, and that everybody signs off on. And that you now have the management behind you to say, no this change control process is a very important thing we must follow it, and anybody who doesn't follow this can then deal with the repercussions of that afterwards. Whenever you're dealing with any type of change, not just for change management or anything, it can be difficult. And your security policies, what you have your back pocket that, says, this is exactly what we've all agreed to, let's follow this going down the road.

**Tags:** certification, change management, comptia, security

**Category:** CompTIA Security+ SY0-401

## **Incident Management – CompTIA Security+ SY0-401: 2.3**

When an incident occurs, there needs to be a clear plan of execution. In this video, you'll learn how to create an incident management strategy for your organization.

Security incidents are going to occur in your environment. The key is how you handle those incidents when they happen. And these could be some very broad types of problems that might occur. It might be somebody hacking into a database and getting all of your customer private information. Maybe a laptop that's stolen or maybe something like a water pipe bursting. And if that occurs in a data center that then becomes a security concern as well. And something that has to be considered in you have to react when that incident happens.

The first thing you should know about mitigating risk here is who you contact, who you talk to inside the organization. Are there organizations or people you have to talk to outside of the organization? This becomes important when you're dealing with finances for instance, if you're a very large financial organization, and a security risk occurs you may be legally obligated to contact government agencies and inform them of that particular security breach. So that's one of first things you have to think about when you're showing up with an incident. Something has just happen is, who do we make aware that this has happened? You also have to think about who's responsible for this problem when an incident occurs. If it's with the database maybe the responsibility of that lies with the database administrator, maybe to your security professionals, maybe it's somebody who is responsible in the data center. So now we get together all of the groups of people responsible for this to resolve or address the incident that has occurred. This is going to be your expert list two, and this may be external people that you're calling.

If somebody's has got into a database, and they've got into very sensitive data, and you really want to go back over this and find out forensically how did they get in here, what happened, we may need knowledge from the outside. We many professionals that deal with this all the time. We're going to call these people that we have a retainer, these people that we've talked to and have a relationship with. Bring them in and get that help if you need it. You also need to think about what the technical steps are going to be, when you arrive when a problem occurs, and you want to make sure that you're able to preserve evidence but still maintain uptime, it's a very interesting balancing act.

If somebody's taken over your email server maybe you unplug the server from your network connection and you start rebuilding locally, because people are going to miss the email if it's down for a few hours. But if this happens to be your primary web server or your primary database, maybe it's not as simple as unplugging this from the internet, maybe we need to provide some access to this database, and instead start working the problem with it connected to the internet. You have to make those decisions sometimes on the fly. And if you have everybody involved, and you're getting feedback from everybody, and everybody can sign up and say, yes that's the proper way we should go for handling this particular incident.

You also have to think about what gets documented. What goes into the report? This information is something that you're going to use to go back over time and find out what happened during that time frame. But it's also something you're going to be able to use in the future. And if you need to be able to have some type of legal action brought against the person that caused this incident then you're going to need to make sure you document as much as possible. So always keep that in mind and always think about documenting exactly what's going on. We have an entire module on incident management. We talk about the things that you're able to do to document pictures, and video, and things that you write down. It becomes an important part of this incident management.

**Tags:** [certification](#), [comptia](#), [incident management](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **User Rights and Permissions – CompTIA Security+ SY0-401: 2.3**

How important is the management of user rights and permissions? In this video, you'll learn which questions you should be asking to provide the best security for your users and your important data.

Another way to mitigate risk is to make sure that people only have access to the resources necessary for them to do their job. Sometimes this access can be a difficult one. Some people may feel they are required more access to the network. You walk into some environments and almost everybody is an administrator on the network. Other environments maybe only one or two people are. Then it creates some problems sometimes. But ultimately remember, it's management that gets to decide who has that level of access. It's you, as the security professional, that must implement that security policy on the network.

You may also want to look at exactly who has access. Just by providing a level of access is important, but if you're providing access to HR data you may want to make sure that just the HR department has access to that data and no other parts of the organization. Again, there will be a group of people that will tell you these are the members of the HR department, these are the only people who should have access to this data, and you have to make sure that both of those things sync up on the network. You also have to think about the type of access people have been granted. If you're in the HR department do you have access to these people's records and can you make changes to those records? And are you keeping track of those changes?

If you think about the management requirements, management may not know the nuances of the types of access that you're able to get on a file server or in a piece of software. It's up to you, as the security professional, to be the liaison here. Be able to understand what management requirements are, and be able to turn those into the technical restrictions you need to just allow the access that's needed. You may also want to audit this every once in while, every few weeks, every few months, or every year. Go back and make sure that all of these requirements are still in place. One example of this is looking at who has administrator access to the network.

When the company was much smaller maybe more people had administration access but now the company is grown, and from a security perspective that's puts us at risk. And we need to make sure that not so many people have access to so much information that might be in our environment. This can be a bit painful if somebody previously was administrator and you now remove that capability from their account, now they feel like they don't have the access that they need. So sometimes you have to work around some of these challenges you have not at a technical level, but at a more personal level with the things that you're putting in place from a security perspective.

**Tags:** [certification](#), [comptia](#), [permissions](#), [security](#), [user rights](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Security Audits – CompTIA Security+ SY0-401: 2.3**

Nobody likes an audit, but it's one of the best things you can do to provide a check of your network security. In this video, you'll learn the importance of an audit and which audit types may be appropriate for your organization.

These audits become really important. We're able to really get down and understand what's going on. But the name is not one we like to hear– somebody's now auditing what we're doing. This double checking becomes an important part of your policies. You really have to police yourself. Sometimes you're bringing in a third-party to look over what you've done, to make sure the things that you put in place are really going to work the way you'd like. And ultimately the idea is that your network's going to be more secure, and you're going to have the right information set the way you really expect them to be. You really have to watch this, because even over a very short period of time, things can change rapidly. A new project is being put in, a very short time frame, maybe some corners are cut relating to security, and now you're going to have to go back and look at what type of security afterwards was provided to all of these people. You also have to think about how often you're going to have a look at these things. You have to spend some time, or set aside some time every month, every few months, every year, to go back and look at what was going on. It's going to be pretty important to see what's going on.

There's certain actions that can be automatically identified, certain things that occur where you can get a message, a red flag, or something that shows up in your logs that says, wait a second. Last week three people were given administrator access. Is that something I want to allow or not?

But you have to have systems in place to be able to look over those things. If you don't have a way to have something automatically go through your logs, or have messages automatically sent to you, it's very possible that all of these checks and balances that you've put in place could be completely missed. And now you have people that have access to the network or to resources that really shouldn't have.

There are different areas of auditing we should focus on. One is a privilege auditing. We've talked a little bit about that. Making sure people have rights and permissions to the areas they should have. And if they should not have those particular rights, make sure they don't have those rights. We've already talked about the different administrators that might be on your network, making sure that people who are administrators or not administrators are configured properly. There's also usage auditing. Are people using the technologies that we have in the proper way? Are people using our internet connection for work purposes or not? Are our systems and our applications on our network configured in a way that are secure? That usage auditing will allow us to see if people are gaining access to these systems or gaining access to these applications and really should not be.

There's also an auditing we should do regarding our processes and procedures, especially for disaster recovery. If we have an incident and we want to manage what happens during that incident, do we have everything in place to be able to make those particular technologies for disaster recovery, for incident management, are all of those processes going to run the way we would expect? So make sure that we audit that and the way that we escalate information during those particular circumstances.

And lastly, our administrative auditing– are we documenting things we should be documenting? Are there places we're missing information? We need to make sure that we're capturing as much as possible that we'll be able to use later on.

**Tags:** administrative, audit, certification, comptia, escalation, privilege, security, usage

**Category:** CompTIA Security+ SY0-401

## **Data Loss and Theft Policies – CompTIA Security+ SY0-401: 2.3**

Our data is some of the most important assets in our organization. In this video, you'll learn about the implementation of data loss and data theft policies.

One of the challenges we have with risk mitigation is making sure that we don't lose resources. If we have data loss, if we have theft, it's becoming a big concern. And it's getting bigger and bigger all the time, because we have more and more data on the network than we've ever had before.

So this is from a physical perspective a relatively easy set of policies to put in place. There's usually processes and procedures. When somebody who is a visitor walks into your building, what are the processes in place? Is there a card lock? Are they able to get in? Do they need a badge? Is it someone we're going to make sure is escorted any time they're inside of our building? There's absolutely things we can put in place to prevent some of these things where people are walking in the door and walking out with a laptop. That should not be occurring. And it's a relatively simple process to put in place.

From a data perspective it becomes a little bit more of a challenge, because it's so easy to carry data around these days. We can copy data to our **MP3 players**, a **USB key**. It's so simple to plug things in, take a **CD-ROM**, walk out of the building. It's in those situations we run into that are sometimes more of a challenge, because it's very, very difficult to watch what's going on.

Maybe we're putting additional policies in place to see when people copy data, when they have access to data, and at least be able to go back over time and figure out what happened when that particular situation occurred. There are threats internally. There are threats externally. We can't just look at one of those. We need to look at both, and make sure that our policies are set up not just for people that we don't know, but also people inside of our environment.

Unfortunately, it's very often the people that we trust inside of our environment that unfortunately are creating problems with loss of data and loss of property. This is, as I mentioned, a bigger and bigger threat every day. We're putting more and more data on our networks, and more and more information. We need to make sure that we have all of the right policies ready to go should any of these types of data loss or physical loss ever occur.

**Tags:** certification, comptia, data loss, data theft, security

**Category:** CompTIA Security+ SY0-401

## **Data Loss Prevention – CompTIA Security+ SY0-401: 2.3**

If there's no monitoring of data leakage, then your customer's private information could find its way out. In this video, you'll learn some strategies around data loss prevention and how some organizations found customer data exfiltrating their network.

The concept of **data loss prevention, or DLP**, is all about making sure that your private data, or your customers' private data, doesn't get outside of your organization or in the hands of people who should not have access to that data. So think about all of this data that these large organizations might have. They'll have credit card numbers. They might have medical or health information. They might have your social security number. All of that information is stored in one or a series of different databases.

The idea is that this information can get out. The bad guys want that info. That is extremely valuable information for them. So if they can get access to your information and gain access to credit card, social security numbers, and other important information, they can take advantage of that. That type of information getting out is called data leakage, and that's what you want to prevent.

And so to prevent that we need to think about all of the different places where this data might be. You have information that is on hard drives and stored in databases. This information is flowing across the network. Some of this information may be on your desktop. This data is in so many different places, moving in so many different ways, that it really is a very broad concept that we have with data loss prevention, because we have to try to prevent that data from getting out in each one of those different areas.

A good example of where you might want **DLP** is this one that dealt with Heartland Payment Systems. Heartland Payment Systems is an organization that processes credit cards. So there's a lot of very, very valuable information there for the bad guys get their hands on. In 2007, unfortunately the bad guys found a SQL injection on one of the Heartland Payment Systems machines, then got access into the Heartland Payment Systems network. And from there, they just sat there and they started gathering information about where they were. The Heartland Payment Systems had no idea that they were in place, had no notification that anything was wrong.

But in 2008, the bad guys started to gather data. They created a way to start capturing packets going across the network. And they took the data that was captured inside those packets and forwarded them off to the bad guys' private servers. And as you can imagine, with a network of this size processing this much information, and the bad guys not even identified as being on the network, they were able to gather a lot of information. Ultimately they gathered over 130 million different transactions. A huge amount of information. And even today, this is one of the largest data breaches that IT has ever seen.

Obviously this speaks to the power and the necessity of being able to watch for data leaking out. And these DLP systems, to be able to look for credit card numbers would have been very valuable if they had been in just the right place to gather the details.

There was a lot to be learned from the breach at Heartland Payment Systems. One of the things they found was that they had all of the right things in place for their network to be PCI compliant. And even with the bad guys in their network fully entrenched, they had run many PCI audits, and had passed them every time.

So what was missing? Well what they realized of course, is the **PCI DSS** requirement is really just a starting point. It's a bare minimum of what you need, because all of this data could be in so many different places. For instance, on your computer, these credit card numbers may come up on the screen, they may be stored locally in a cache. We call this data in use. And you would need some type of endpoint data loss prevention system to be able to look for this data and make sure that the person who had access to that on their desktop was really only getting access to what they needed.

Another type of DLP system is one that's on your network. We often refer to this as data in motion, because as those packets are going back and forth, they're not really stored anywhere, they're just being moved around the network. So there are a number of network-based DLP systems that can look for things like credit cards inside of the packets themselves. Look for a specific string of text that's inside of those packets, and if it ever sees those or maybe a certain number of them over a certain amount of time, they can inform you that is going across your network.

Another type of data loss prevention system is on your server, where the data is stored on the hard drive or in a database. We call that **Data at Rest**. And that type of DLP system is able to identify when that information is moved or placed onto a hard drive and stored there for any amount of time.

If you want to try some of these DLP systems yourself, you could go out and grab an open source version of something called **MyDLP**— this is a community edition. It allows you to install DLP on a machine, or a DLP piece of software on a machine. And you can look for data going across your network.

It's these types of data loss prevention systems that are becoming much more important these days as the data on our networks becomes much more valuable.

**Tags:** [certification](#), [comptia](#), [data loss prevention](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Order of Volatility – CompTIA Security+ SY0-401: 2.4**

Not all data sticks around, and some data stays around longer than others. In this video, you'll learn about the order of data volatility and which data should be gathered more urgently than others.

A big part of incident response is dealing with intrusions, dealing with incidents, and specifically how you deal with those from a forensics level. Forensics is talking about the collection and the protection of the information that you're going to gather when one of these incidents occur. There are data sources that you get from many different places—not just on a computer, not just on the network, not just from notes that you take. There's a combination of a lot of different places you go to gather this information, and different things you can do to help protect your network and protect the organization should one of these incidents occur.

If you'd like a nice overview of some of these forensics methodologies, there's an **RFC 3227**. Google that. It's called **Guidelines for Evidence Collection and Archiving**. And it's a good set of best practices. Very high level on some of the things that you need to keep in mind when you're collecting this type of evidence after an incident has occurred.

There is a standard for digital forensics. And digital forensics itself could really be an entirely separate training course in itself. There's so much involved with digital forensics, but the basic process means that you acquire, you analyze, and you report. Those three things are the watch words for digital forensics. Those are the things that you keep in mind.

We're going to talk about acquisition analysis and reporting in this and the next video as we talk about forensics. The details of forensics are very important. You need to get in and look for everything and anything. You need to know how to look for this information, and what to look for. And you have to be someone who takes a lot of notes, a lot of very detailed notes. Sometimes the things that you write down and the information that you gather may not even seem that important when you're doing it, but later on when you start piecing everything together, you'll find that these notes that you've made may be very, very important to putting everything together.

In forensics there's the concept of the volatility of data. And when you're collecting evidence, there is an order of volatility that you want to follow. The volatility of data refers to how long the data is going to stick around—how long is this information going to be here before it's not available for us to see anymore. That's one of the challenges with digital forensics is that these bits and bytes are very electrical. In some cases, they may be gone in a matter of nanoseconds. Other cases, they may be around for much longer time frame.

So the idea is that you gather the most volatile data first—the data that has the potential for disappearing the most is what you want to gather very first thing. The data that could be around for a longer period of time, you at least have a little bit of time that you could wait before you have to gather that data before it disappears. So this order of volatility becomes very important.

So what's volatile and what isn't? When you look at data like we have, information that might be in the registers or in your processor cache on your computer is around for a matter of nanoseconds. These registers are changing all the time. That would certainly be very volatile data. If we could take a snapshot of our registers and of our cache, that

snapshot's going to be different nanoseconds later. So that's one that is extremely volatile.

Next volatile on our list here— these are some examples. This is obviously not a comprehensive list, but things like a routing table and ARP cache, kernel statistics, information that's in the normal memory of your computer. Those would be a little less volatile than things that are in your register.

Next down, temporary file systems. Those tend to be around for a little bit of time. But being a temporary file system, they tend to be written over eventually, sometimes that's seconds later, sometimes that's minutes later.

Next is disk. When we store something to disk, that's generally something that's going to be there for a while. Unfortunately of course, things could come along and erase or write over that data, so there still is a volatility associated with it. If we catch it at a certain point though, there's a pretty good chance we're going to be able to see what's there.

Remote logging and monitoring data. If there's information that went through a firewall, there are logs in a router or a switch, all of those logs may be written somewhere. The problem is that on most of these systems, their logs eventually over write themselves. Sometimes that's a day later. Sometimes that's a week later. Sometimes it's an hour later. But generally we think of those as being less volatile than something that might be on someone's hard drive.

The network topology and physical configuration of a system. That again is a little bit less volatile than some logs you might have. And down here at the bottom, archival media. A **DVD ROM, a CD ROM**, something that's stored on tape somewhere and archived and sent somewhere else— probably we can have as one of the least volatile data sources you can find, because it's unlikely that that particular digital information is going to change any time in the near future.

**Tags:** certification, comptia, free, james messer, order of volatility, professor messer, security

**Category:** CompTIA Security+ SY0-401

### **Capturing System Images – CompTIA Security+ SY0-401: 2.4**

Imaging a system can be a very effective way of preserving evidence. In this video, you'll learn about system imaging and some strategies for obtaining system data without directly imaging a storage drive.

If we're collecting data from a hard drive or from a digital storage media, we may want to grab an image of that drive. We would like an exact representation of that drive that we could copy off somewhere else. And if we ever needed to reference what was on that drive, we have an exact duplicate in time frozen, that we would be able to see what's going on.

Now when you start looking in forensics of hard drives, there's a lot of information on the drive. Some of the information on the drive has been deleted, but as you know, when you delete files from a hard drive, you aren't actually deleting the file, you're simply removing a pointer to the file. So having an exact duplicate of the file, one that you'd like to create on what we usually refer to as **a bit-for-bit copy**. Sometimes it's called a **byte-for-byte copy**, which means we're going to every bit on the drive, and we're copying exactly the contents of that drive from one drive to another.

We aren't copying the files. We aren't just saying copy file from point A to point B. We're saying copy every single bit on that drive and duplicate that bit on an image that we're creating. We want to get every single bit of data we possibly can get there. And often

when you're going back and trying to understand what happened, if somebody deleted some files but they didn't actually do a secure delete, the information you need might still be on that drive. So you want to be in a position that you could recover that information if at all possible.

Usually you have some software— a boot **DVD**, a specific **LINUX** image that you could use to boot up some imaging software that can do this. There are a number of forensics imaging, **live CDs, or live DVDs**, that you could boot a system from and do some of these forensic system captures.

Now sometimes this bootable device isn't available, or doing it by software perhaps isn't the best way go about it. You'd like to remove the disk from the system itself and plug it into a device that will allow you to do a copy of that disk, but ensure that nothing could ever write to the disk. And to do that, you would need one of these hardware-based blockers. What you do is plug your drive in, and it's a one-way device. It's a one-way bridge that will allow you to read anything you would like from that media.

Some of these are pretty complex. It can be a standard hard drive, **SATA, IDE, SCSI, USB**. You plug-in some media into that, and you can only read. It is impossible to write to that media. These are specifically built, as the name says, as forensics. This is so you can be assured that you would not be modifying anything on that particular drive.

There's some of these that are built to be internal to a system, some that are built to be external to a system. This happens to be one for **ESAT— external SATA connections**— so that you can then plug-in firewire and SATA connections, and have that data roll off to other disks. It's very portable in that way.

If there are backup tapes, or backup images, or backup systems from a computer, make sure you get those as well. Don't forget there might be some really good data from what was on this machine stored somewhere else. And if they're very good at doing backups, that could actually help you in doing some forensics and understand what did this system have on it yesterday, the day before, or last week, or even further, depending on how many backup tapes you happen to have, or backup images you happen to have.

So all of these sources are going to be very useful to you, especially if you're gathering information and wanting to recreate what was on that particular hard drive.

**Tags:** certification, comptia, imaging, security, system image

**Category:** CompTIA Security+ SY0-401

### **Capturing Network Traffic and Logs – CompTIA Security+ SY0-401: 2.4**

The packets traversing network can be a wealth of information. In this video, you'll learn how to collect information from network traffic and logs.

A lot of the things we do with our computers traverse the network. They're sending information back and forth over our network connections. And being able to pull up traffic logs can tell you a lot about what a person did, where they visited on the network, what information might have been transferred. These are very, very common to find. And things like firewalls— which are designed to be security devices, and they're designed to protect your network— very often those are logging everything, every flow of data going in and out. At least at a high level. And it will tell you that this particular IP address and perhaps even this user went to the internet, and they transferred a file to this server. It may not show you the contents of the file, but at least you know that's what happened. And then you can at least go from there and determine what file did you transfer? Maybe that file's on someone's hard drive. Maybe you can go to the server they transferred it to and obtain the file that way.

The firewalls usually have a great deal of detail there. Things like switches and routers don't log a lot of user-level information generally. They're telling you that a port on the

switch was activated and not activated. But not necessarily what traffic traversed the switch or router. So if you're a security person, you're looking for a lot more detail, the firewall may be the place you go from a network logging perspective.

There's also a lot of log stored in intrusion detection and intrusion prevention systems. Normally they're just logging unusual traffic or traffic that happened to match a particular signature. So not a comprehensive set. But if somebody's downloading a file that happens to have in it some information and something that fires off this signature— they're downloading a vulnerability scanner, they're downloading some code they can use to attack another machine— your IDS or your IPS might identify that. Yet another data source to go to.

And one of the ultimate data sources, if you have this luxury, is the way to go back to the raw network traffic that traversed the network. Not everybody has this capability, but there are stream-to-disk solutions— that's what this is called— that takes every bit and byte going in and out of your organization or past a network connection, and it stores it on this massive array, these terabytes and terabytes and terabytes of hard drive space.

These are usually recording every single bit and byte. And so this is great. You can now go back in time and pull out the exact information that traversed the network. Which means you can rebuild emails that went back and forth. You can recreate the files that were transferred. You can see the exact page in a browser when somebody visited a website. You can see everything, because it's all in that data going over the network. And if you have a stream-to-disk solution, that's a great place to go to help recreate exactly what went across the network.

**Tags:** certification, comptia, logs, network, packets, security, traffic

**Category:** CompTIA Security+ SY0-401

### **Capturing Video – CompTIA Security+ SY0-401: 2.4**

A video can sometimes provide much more information than weeks of log files. In this video, you'll learn some strategies for collecting evidence with video.

Another good source of forensics data is video. This can be video that's internal, on the computer itself. It can be video that's external of the computer, or the network, or the place where the particular event occurred. This gives you a moving record of what went on. Sometimes if you're approaching a scene of an incident, you may want to start recording yourself. Turn on your own video. It's so easy today. You've got these mobile video devices. Our phones these days are HD recorders— perfect place to go to record exactly what you find in the state you find, that you can then share with other people. That way you've got a step-by-step, second-by-second now archive of exactly what you did. And this might be recording what's on a computer screen, recording the situation around a data center, understanding if a door was jimmied open, you may be able to get a picture— a video— of exactly what that looked like, and really have more information available to you.

Coming up to a screen that has been compromised and recording it with an external camera means that you're able to at least see what's going on without affecting anything else going on on the computer system or the network. And having those mobile devices that you carry with you all the time just gives you another place to go there. Don't forget your security cameras. If you have surveillance systems and security cameras, those sometimes have a data they'll store over time, but they're still volatile. You want to be sure you're able to catch those before they overwrite themselves. And maybe sometimes you're able to see people going in and out of the building. Even if it didn't capture the exact incident where it occurred, you've got cameras elsewhere that you'll be able to help with the evidence and the information that you're gathering together.

You also of course have to archive this video content. It's still evidence. It's things that you want to keep. This may indeed be some of the most important information you have, just because of the audio and the video context associated with it. So make sure you archive it in a way that later on, months later, a year later, when you need to go back to it, these things tend to drag out over long periods of time, make sure it's something that will be accessible to you, and something that will be in a format that you'll be able to view.

**Tags:** [certification](#), [comptia](#), [security](#), [video](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Recording Time Offsets – CompTIA Security+ SY0-401: 2.4**

Time is relative, and that's certainly the case with a computer's internal clock. In this video, you'll learn how operating systems count time and how a computer's timezone setting can change how you look at data.

So what time is it? Well statistically speaking, you're probably in a different time zone than me. You're certainly watching this video at a different time than I recorded it. There's a difference there. And so when you're dealing with time, you have to be very precise and very specific about when you saw something, how you saw it, when a file was saved, when a file was accessed.

One of the things we want to look at are the offsets of time. Now if this is the Windows operating system for instance, Windows uses a **64-bit time stamp**. And it's a time stamp that counts the number of 100 nanosecond intervals that have occurred since January 1 of the year 1601 at 00:00:00 GMT. Obviously, this means that it's going to stop working in about— oh, I don't know— 58,000 years. So if anything, Microsoft was thinking ahead of the game here. You aren't going to run out of time any time in the next 58,000 years. Maybe by then, we'll have a different time stamp we can work with.

In **Unix, or Linux**, we have a **32-bit time stamp** to deal with. This recognizes the number of seconds that have occurred since January 1 of 1970 at zero GMT. Now notice that this is a 32-bit time stamp, which means it's going to stop working soon. We don't have that many numbers or seconds that we can deal with here. So it's going to stop working on Monday, December 2 of 2030.

Now if you were around for Y2K, this will be the next one we have to deal with. Y2K plus 30, I guess you can call it. At 7:00 PM, 7:42:58. 19:42:58 GMT. So there's your challenge there, is that the timestamps notice between Windows and Linux or Unix based systems— POSIX-based systems— very different in the way that they count what time it is.

And when you sit down in front of the computer and you're looking at the file and the timestamps, there's also differences on how the operating system is storing those. And different file systems store timestamps differently, and **file allocation table**— the **FAT table**— time is stored in local time. That means that whatever the local time is on your computer. If it's five in the afternoon your time, it stores it as five in the afternoon. You have to keep that in mind when you're looking at the time stamps.

If you're using **NTFS**, the time is stored as GMT. And your operating system changes the time on the fly to show you what your local time is, but the reality is the time stamp of the file is in GMT. So that's another thing to keep track of from a forensics perspective.

You also when you sit down in front of these computers, then you have to know what time zone is configured on the computer, so that when you're looking at a screen shot of timestamps, you understand relative to GMT what the actual time is. So you want to look at the Windows registry, which is the ultimate source of where this is stored in Windows. And there are many different values in the registry, because you can set a time, you can set what your time zone is, you can set whether you're going to have daylight saving time

take effect there, whether it's going to time change information automatically or not. There's a lot to the time.

So by storing this and looking at the time, the clock on your computer, you also need to understand what the offset is set to, and then you'll be able to have everything go back to one relative time stamp. Usually GMT is one that we're very commonly using as a standard relative time stamp. But you can see here now how important it is when you're collecting data to make sure you have the correct time and you understand what the time offset is of that computer.

**Tags:** certification, comptia, offset, security, time, timezone

**Category:** CompTIA Security+ SY0-401

### **Taking Hashes – CompTIA Security+ SY0-401: 2.4**

How can you tell if a digital file on a piece of storage media has been altered? In this video, you'll learn about hashing, and I'll demonstrate how to compare hashes with a file that has been altered.

One of the challenges with the digital technology is it's very difficult to tell if your digital file has been changed. You can't just pick up a disk or CD-ROM and look at it with the human eye and try to determine if something's changed on that file between yesterday and today. It's one of the challenges we have.

So of course we have some things we can do to check files. And the easiest thing we can do, and probably one of the most accepted, is doing something like taking a hash of the file. This allows us to essentially create a fingerprint of a file. In fact, you'll see it referred to as a digital fingerprint. And if that file or any part of that file ever changes, you'll notice that the fingerprint will change. We're going to look at this in just a moment.

The normal, or one of the most common ways to do this, is with something called **Message Digest 5**. And you'll see it always abbreviated as **MD5**. Oh, you need to generate an **MD5 hash** of that file. What's the MD5 of that file? You may have also seen this if you've been downloading files from the internet, and they have a download link. And right next to the download link, they have a big long hex string, and they say, this is the MD5 of the file. The idea being is that the file was put on the website, and the fingerprint was made. When you download it, you can see if the fingerprint is the same as what it was on that web page. And at least you can see if those things sync up properly.

This **MD5 hash** is 128 bits long. As I mentioned, it is displayed as a hexadecimal string. There is an interesting chance of duplication— 1 in 2 to the 128th power. So we're talking about 230 billion, billion, billion, billion of a chance that a change to a file would be exactly the same as the fingerprint that was made originally.

So one of the things we can say is, it's pretty impossible, or relatively impossible, to have some modification to a file and have the fingerprint turn out to be exactly the same. In reality, you're never going to hit this. The statistical odds are staggering.

Another type of hash is a **CRC**, or **cyclical redundancy check**. This is a much smaller type of check. It's only **32 bits long**. Again, it is displayed as hexadecimal. And you can see the chances here of having a CRC duplicated after a change. 1 in 2 to the 32nd second power, which is just over 4 billion to one.

Now one of the things that you'll notice is that in your hardware, maybe hard drive checks and memory checks, they use CRC's. And that's a really good way to use that particular technology, because it can be calculated relatively fast, and it's something that you can see very, very quickly as it goes by.

An MD5, because it is 128 bytes long, it takes a little bit longer. There's a few more calculations that have to take place to be able to create that fingerprint. So you'll see most of the time when we're trying to verify that a file or an image is exactly the same as when we left it, an MD5 is really the one that is going to give us that flexibility.

You'll very often see CRC's when you're looking at how hard drives are writing and checking their information. But rarely do we use a CRC from a security forensic standpoint. The idea is that we would check a file, an image, any digital piece of information, we can create an MD5 fingerprint on that. And that allows us to check it after the fact or verify it anytime during this file processing. We might take the file off of the computer, move it to another media. We might copy the image of a hard drive and move it somewhere else. We can check the fingerprint every step along the way.

So just like taking regular fingerprints at a crime scene, you will absolutely get a list of a bunch of fingerprints when you're doing this type of incident response to make sure later on down the line— a day later, a week later, a year later— that you've got exactly the same file you started with. And your fingerprint is a very, very good way to do that.

Let's see one of these hashes in action I'm running on my Mac desktop here, but you could be on Windows. There's many utilities you can use to create MD5 on Windows, and LINUX, and other operating systems as well. I have a single file on my hard drive called evidence.txt. And in the Mac OS X, I can simply do MD5, and the name of the file, and it will give me this hexadecimal representation of evidence.txt.

So I know I can take that information, log it, and make sure that I have it, and that way if I want to check this file later, I can confirm that that evidence.txt MD5 fingerprint matches exactly what's here. But what if somebody changes the file? Let's change this file evidence.txt and say this file does not have important information. And let's save that. If I run the same MD5 check now, notice the fingerprint's very different. I made modifications to the file. And if you went back later and you said, wait a second, this evidence.txt MD5 hash does not match the one that was made originally, then you know this file is not the original file. Something has been modified. Now you have to figure out why something changed between the time when you grabbed the file and the time that you have it right now.

**Tags:** [certification](#), [comptia](#), [hash](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Taking Screenshots – CompTIA Security+ SY0-401: 2.4**

There are many ways to capture the information that appears on a screen. In this video, you'll learn a few techniques for taking screenshots across many different operating systems.

Screenshots can be another useful way to grab what's on the screen. Sometimes it's difficult to reproduce what you may run into when you first walk up to a computer. You see something on the screen that looks suspicious. If you were to grab an image of the drive, that allows you to get the information that was on that computer, but it doesn't necessarily allow you to reproduce what's happening on the screen.

And one of the challenges is, how do you take a snapshot of what's on the screen without disturbing the computer? Obviously you can hit the Print Screen command in the Mac operating system, in the LINUX operating systems, there's other utilities and other ways to grab screen information, but then you're changing what's on that computer. And you may not have the ability to do that. So sometimes it's not really the easiest process.

But there are things you can do. You can of course capture it externally. One of the nice things about our latest generation of mobile devices is we can pull out a phone, and all of

our phones tend to have cameras on them these days. And they tend to have some very high resolution cameras with some really nice photo capabilities.

Pull out your camera, take a snapshot of what you see on the screen right then. Capture it. Then you'll have error information. You'll have a message of what people saw on their screen. You'll have what was on an email that somebody was typing but did not send, and may be difficult to reproduce later. All of that will be on the screen and you'll capture it to your phone.

You also have some functions within the operating system itself. In Windows you can hit the Print Screen key. And now you have the entire screen in your clipboard. There's other third-party utilities you can get as well on any operating system that can grab what's on the screen and save it to a file. There's some nice forensics capabilities, some forensics programs, on USB drives where you can plug in the USB key, run the executable from there, and it will save the file to the USB key, thereby minimizing the impact to the hard drive of that computer itself.

**Tags:** certification, comptia, screenshot, security

**Category:** CompTIA Security+ SY0-401

### **Interviewing Witnesses – CompTIA Security+ SY0-401: 2.4**

There's sometimes no replacement for having a first-hand account of an incident. In this video, you'll learn the advantages and disadvantages of interviewing witnesses.

A useful non-digital source of information is **witnesses**— finding people who have seen something going on. They saw a co-worker acting peculiar. Maybe they saw someone walk in the front door that they didn't recognize. And you want to find out more about what they may have seen. So document, talk to them, call them, write down what they've seen, gather information about what they happen to have identified that day. And try to get very quickly to that person. The more time goes by, the more we tend to forget. So you want to get something that's very top of mind, very fresh in their memory.

Now keep in mind of course, that witnesses are not 100% accurate. They're giving you an idea of what they think they saw. Sometimes what we think we see is not actually what occurred. But in our mind, that's absolutely what happened, even if it isn't. So although the information they're giving you they may feel is absolutely 100% accurate, you should always find ways to get a second party to verify what they're doing. Or in some way try to figure out what they're saying, and try to put it in context of what actually occurred.

**Tags:** certification, comptia, security, witness

**Category:** CompTIA Security+ SY0-401

## **Tracking Man-Hours and Expenses – CompTIA Security+ SY0-401: 2.4**

Processing a security incident can be an expensive endeavor. In this video, you'll learn the importance of proper expense accounting and why tracking these details is an important consideration.

And after tracking these people, and getting your interviews, and finding out what's happened with an incident, you're going to find that the cost of these go up, and up, and up. You may have seen sometimes when you're reading some of our trade magazines, that a company said that we had a breach, and it cost this much money. And what they've done is compiled not just the financial impact of that event itself, but they've also calculated in how long did it take us to research this? How many people were involved? How many man-hours were spent on this? All of these resources cost money. So you're going to have to see not only what happened immediately when the event occurred, but as time goes on, as you're researching and trying to recover from that incident, how long did it really take.

And some of these things go into our legal system. They can take months or even years. And of course, that all means more man-hours are put towards resolving this issue. These can have an impact on the bottom line. Somebody can steal money from your organization. That's obviously something that would impact what resources your company might have. But that bottom line view can be also very wide-ranging. You also have people that are spending time on the back end, trying to piece together what happened, that could be doing other things that would make your organization more money. So you have to do some interesting calculations over these man-hours and what you're doing.

Try to be as accurate as possible, because if you find the person who created this financial issue for your organization, they may be required to pay you restitution. So you may have to provide the courts with exactly how much money we believe this particular incident cost us, and we're going to have someone pay us back for all of this time.

**Tags:** [certification](#), [comptia](#), [expenses](#), [man-hours](#), [security](#), [tracking](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Chain of Custody – CompTIA Security+ SY0-401: 2.4**

If you're handling evidence, then you'll need to follow a strict chain of custody. In this video, you'll learn about chain of custody and how it can assist with the resolution of security incidents.

In many situations where you are collecting evidence, you have to maintain a chain of custody. That means that the integrity of what was gathered at the crime scene or the incident scene is something that you can later on look at and verify that what you gathered during that initial phase is exactly what you're looking at later. It's important internally of course, and incredibly important when you get into legalities and being able to prosecute people for bad things that they've done to your resources or to your environment.

So anyone who contacts, runs into this evidence, that touches it, moves it, transports it, does anything with it, generally has to be part of the chain of custody. The idea here is we're preventing any of this from being tampered. Many of these evidence bags have a section at the top where you can seal it. There is no way to get into this bag unless you tear this bag apart. And that again, now you sign off on it. Yes, I opened the bag. If you now need to reseal it, you have to use a separate bag. And normally you seal everything—the original bag and your evidence along with that bag.

You're going to label everything. You're going to catalog everything. You're going to take pictures of as much as possible. You're going to seal it and probably store it away for at least for a temporary amount of time. As I mentioned earlier, these things can tend to go on a number of days, a number of weeks, a number of years, if this is going into our legal

system. So that you need to be able to pull this out a year later and determine has this been tampered with, is this exactly the same as what I put in this box a year ago. You have to maintain that chain of custody to be able to do that.

**Tags:** certification, chain of custody, comptia, security

**Category:** CompTIA Security+ SY0-401

### **Big Data Analysis – CompTIA Security+ SY0-401: 2.4**

The era of big data has arrived, and we're now faced with analyzing more data than ever before. In this video, you'll learn about big data analysis and the techniques that we're using to sift through our huge data stores.

The term big data is much more than what just the name implies. Obviously, we are talking about a lot of data. Thus the name big data. But the idea behind big data goes so much farther, beyond just a large collection of data. This data is something that we are collecting from many different places and we're storing it. We have such facilities these days to store large amounts of information. That we're storing this data, we don't even know if the data is correlated with each other in any way. And in fact, many of the tools that we use to analyze the data aren't even able to take into account this massive amount of data. This is truly an emerging type of technology and one that will provide us with some interesting insights into information.

From a network security perspective when we think about big data we're really thinking about collecting data from many different kinds of devices. These are things that you might traditionally think of like, firewalls or intrusion prevention systems. But certainly there's information to be gathered from switches, and routers, perhaps, even file servers. What if we collected all of the internet searches that were done inside of your organization? What if every URL was categorized and stored? You can imagine the amount of data that you would collect, just over a single day, would be truly massive.

Not only do we have a challenge with collecting this massive amount of data in one place, but we also the challenge understanding of the different log types that we're gathering. We're gathering information from firewalls and that data is very different than what we might gather from a file server. And of course, all of that is different than what we might gather from internet search results. All of this data somehow has to be gathered together, and analyzed, and that's where the real secret of big data comes in. We're getting much better at storing this data. We're now starting to have much better effect at going through the data and analyzing what we've collected. The data stores are so large, and the amount of data that we have to query is so diverse that we really can't use their traditional forms of data querying that we've used in the past. We have to come up with new ways to be able to correlate this information together and provide some visualization of what exists inside that data store.

One of the most intriguing parts of big data is you really don't know what you have until you start putting the data together and looking at it in different ways. Very often, correlations of data may appear that are completely unknown prior to that point. So it's very important that the query tools that we have allow us to view the data in different ways. Maybe we'd like to see graphs of correlations between data. Perhaps, perform some statistical analysis that might lead us down a particular path, or perhaps things like tag clouds. They can take certain correlations and make them much larger, and have them pop more towards the top. This type of analysis is going to allow security professionals to look at information across the entire enterprise in ways that they never have before.

**Tags:** analysis, big data, certification, comptia, security

**Category:** CompTIA Security+ SY0-401

## **Preparing for an Incident – CompTIA Security+ SY0-401: 2.5**

What happens when there's an incident? In this video, you'll learn about the industry best practices and some strategies for preparing for (and perhaps preventing) an incident.

When a security incident occurs it's very important to have a set of processes and procedures in place. This way you'll be able to know exactly what to do to get your systems back up and running again. And you'll also know what you can do to help avoid this problem in the future.

One of the first significant steps after a security incident has occurred is to get your systems back up and running again. This may be something as simple as restoring access to the device. But of course this also may resolve cleaning out any issues that still may be on those systems due to this security incident. So it may require bringing in some specialization or people that have gone through the resolution of these problems before.

Why you're restoring your systems you want to be very careful not to remove any evidence. If this is a malicious incident and you want to prosecute this person you're going to need to have as much evidence as possible. You also want to understand exactly how this incident occurred. By analyzing this you'll have a better idea of how to prevent these incidents in the future. This may be something relatively simple, like updating an operating system or patching an application, or it may require a large change to processes and procedures in your organization. If you'd like to read more on how to handle computer security incidents you can go to the **National Institute of Standards and Technology** website and search for **NIST Special Publication 800-61**. This is the computer security incident handling guide and it gives you a process and procedures of how to handle these particular incidents. It starts with the preparation before an incident occurs, goes through the detection and analysis phase, eradicate and recover your systems, and ultimately what you can do after the incident has occurred.

The preparation process is critical to handling these incidents. You want to be sure that you have every communication method available to you, and the ways that people need to be contacted. So you should already have a contact list for everybody who will be handling this incident. And you want to be sure that you have the proper methods in place, so that you can contact everybody on your list.

During the incident analysis you may need specialized hardware and software to be able to understand what occurred during that incident. So you may need a specialized laptop, you may need your own cameras to be able to photograph and capture information that occurred during the incident, or you may need specialized software that will allow you to do forensics or perform disk images of hard drives. These incidents might occur anywhere in your organization so you need to have as much documentation, as possible, so that you understand where hardware may be located or what the network diagram might look like. Might be useful, as well, to understand some baselines or have some critical file hashes, so that you can compare a before and after and understand if any changes occurred during that security incident.

If you're cleaning up after the incident you may need to completely wipe the slate and begin fresh. So it's also nice to have installation media for your operating systems, or to have images of applications, or pre-built systems. That way you can get up and running as quickly as possible. And ultimately, there should be a set of policies and procedures. Everyone should have a set of jobs to do, and everyone should understand, exactly, what needs to be accomplished to resolve this particular security incident. Of course, the best possible scenario is to avoid the incident entirely. So it might be useful to go through some preventive steps that you can use to help prevent these incidents from occurring in the first place. One of these might be performing a risk assessment. This is something you can do periodically to understand if all of your systems are properly patched, and understand if there are any security issues associated with the devices, the hardware, and the software in your organization.

Operating systems can be especially vulnerable. So you may want to have some documentation and procedures on how to harden the operating system you use in your environment. You also want to be sure that you're updating this operating system with security patches. And of course, monitor the operating system to see if you can notice any anomalies with the operation of that system.

From a network security perspective, you should have hardware and software in place to be able to protect the flows of traffic going through your network, and to analyze those flows to see if there might be anything malicious inside of that. It's very common to have a firewall, to check for traffic, that may be traversing two different networks. Virtual private networks can be used if people are connecting from outside of the network. And intrusion prevention systems can stop the malicious software directly on the network, so that it never reaches the end users. It's also very common to run anti-malware software on our end point devices. That way if any malware does find its way to the desktop, we can stop it from executing in the operating system itself. It's also common to do this in our operating systems, running on our file servers and our email servers, so that we can stop the malware in these central infrastructure devices. And of course, we want to train our users and make sure they know exactly what the latest security techniques might be. By using all of these methods we cannot only protect when a security incident has occurred, but we may be able to prevent one from ever occurring in the first place.

**Tags:** certification, comptia, incident, prepare, security

**Category:** CompTIA Security+ SY0-401

### **Incident Identification – CompTIA Security+ SY0-401: 2.5**

One of the challenges of security incidents is recognizing that one has occurred. In this video, you'll learn about some techniques that can help you detect incidents and attacks.

Detecting a security incident is not the easiest thing to do. These incidents take many shapes and forms. They may have a different amount of detail. They may attacked different kinds of systems. You're never quite certain exactly where the incident is going to come from. One of the challenges we have is that our networks are constantly under attack. If you're connected to the internet there's automated processes, bots, worms, and people maliciously trying to gain access to your systems. So the question really is, of this traffic, how much of it is a legitimate threat, and how much of it is going to be stopped by the existing systems that we have in place?

It's multiple layers of security that we have to spread across every part of our infrastructure, and it really requires some specialized knowledge and tools to be able to accomplish that. It would be really great if we could identify that an incident was going to occur before it actually did occur. And one way to do that is to look at different pieces of our network to understand where these changes, or significant precursors, may be happening. One place to gather of these precursors is in something like a Web server log. You can look at what people are hitting your web server, and you can also see when different devices, or different scripts, may be running automated vulnerability checks against your servers, and this may give you a heads up at somebody's trying to gain access into your systems.

Another precursor may be the very common monthly announcements vulnerabilities. Microsoft, for instance, announces vulnerabilities, and almost immediately, you see the bad guys trying to take advantage of these open holes in your operating systems, before you have a chance to patch them. So you may see an increase in the number of people trying to use those vulnerabilities, against the servers that you already have in place. And some precursors may be very obvious. The bad guys might contact you directly and say,

that you need to pay them a certain amount of money, or they're going to hit devices with a massive denial of service attack. In this particular case you have a decision to make on whether you pay them their money, or you protect your systems against these kinds of security incidents.

There's a number of things that you can monitor to see if an attacker might be under way. This obviously is going to be important, because the sooner we can stop the attack, the less damage is going to be occurring in your network. One way is to look for things like buffer overflow attempts. Which is a very common way to take advantage of bad software in an application or an operating system. You can look to intrusion prevention systems to provide you with information about things like buffer overflows, because they tend to have signatures that are specifically designed to look for these on the network and inside of an operating system. On the devices themselves you can constantly update your anti-virus and your anti-malware signatures, so that they may also identify malicious software that may try to execute in the operating system itself. In those cases the software also generally contacts you, so you'll have an idea of where these particular attack vectors are coming from.

It's also very common in your file servers, where the operating system files, in the documents don't change very often to lock them down and then monitor the files. That way if somebody does gain access into the operating system and they try to modify what exists, you'll be notified that somebody's made a change to these files that normally would never be changing. And another way to look for any type of security incidents that may be occurring is to look at your network traffic. Network tends to be very predictable day, after, day, after, day. If there are any significant changes, or things deviate, from what is normal it may be indicative of an ongoing problem. Hopefully, these precursors, and these active indicators, of a security incident can help you get a better idea of what might be happening in your environment.

**Tags:** certification, comptia, detect, identification, incident, security

**Category:** CompTIA Security+ SY0-401

### **Incident Escalation and Notification – CompTIA Security+ SY0-401: 2.5**

When you're involved with an incident, communication is key. In this video, you'll learn about escalation procedures and strategies for security notifications.

When an incident is occurring there are a number of people that will need to be kept in the loop and notified as to what's happening with the progress. Whenever we look at internally in an organization it's very common to notify people like the **CIO** or the **Head of Information Security**, and certainly the teams that are responsible for responding to these kinds of incidents. We may also need to go outside of the information technology group to inform people like human resources or our public affairs group, and certainly the legal department may need to be notified as to what's occurring. You may need even go outside the organization. In some cases, this may be a criminal act and you may need to contact your local law enforcement. And for government agencies there may be requirement to contact the US cert organization.

During a security incident the ability to communicate with others is incredibly important. Everyone needs to be up to date and informed of exactly what's going on, especially if multiple people are working on different aspects of the security incident. So there's different ways we should consider communicating, whenever one of these things occurs.

We can of course communicate via email, if your email is working properly. In fact, you may want to have a secondary email systems in place in case your internal email system

is part of the security incident. You may also want to go to the web and post information on a public or a private internal web page that way people can get updated immediately on what the latest status might be. Of course, communicating by voice is a great way to do this. So you'll need to have your contact list and understand exactly who you should be communicating with over the phone. If you have the ability for everybody to get into a room you may want to have in person updates, or have periodic meetings where everybody can sit in a room for 15 or 30 minutes and discuss where they are with resolving this particular security incident. Sometimes you may want to have this set up on an automated voicemail system so that people can call a centralized phone number and get a status of where things are with the security incident. And you may get rid of technology completely and do everything by paper, have a centralized board and you compose notices, and leave messages on that centralized board, and avoid any type of technology, whatsoever. Information Exchange is an incredibly important part of an incident response. So you want to be sure that you have all of these methods in place if you ever have to respond to some type of security incident.

**Tags:** certification, comptia, escalation, incident, notification, security

**Category:** CompTIA Security+ SY0-401

### **Incident Mitigation and Isolation – CompTIA Security+ SY0-401: 2.5**

There are many ways to limit the impact of a security incident. In this video, you'll learn about methods to stop the attack and limit the scope of the damage

The objective of an incident mitigation is to limit the scope of what the attack might do your systems. If there's any damage, or any widening of this incident, we may be able to contain it, and limit it from going outside the scope of that containment. These mitigation options might be very different depending on what kind of attack it is. For instance, if their attacking your internal email system you may be able to mitigate that by removing the email system completely from your network. But if the attack is going after your publicly facing web services you may have a limited number of options available to you. Hopefully, then you can go back to your planning process and examine what options may be available to you. Generally, you want to have your critical resources available, as long as possible, you don't want to remove your publicly facing web servers from the internet, if you can continue to run and still yet contain that particular security incident. The goal is to collect as much information as possible. And then you can use all of that information to help make your decisions on the best way to mitigate this issue.

So what kind of criteria should you consider when you're planning which strategy to follow, to mitigate this particular security incident? Well, one thing that you can consider is how much damage may be actively occurring, or how much data may be leaving your environment. If you can see the data leaving your network and going outside of your environment, or you can see that active pages on your website are being destroyed, erased, or changed, then you might want to very quickly begin some type of aggressive mitigation.

When these incidents are under way you want to gather as much evidence as possible. Especially, if this is happening from a third party and it may be a criminal attack, you may want to have this available. Especially, if this goes into a trial or any type of legal process. You can refer back to the video on planning for security incident, where we describe all of the things that you can use to help gather evidence when these types of problems occur. You should also consider how your mitigation might affect the services that are available to your customers, or the resources that are available inside of your network to the rest of the organization. The best possible scenario would be to mitigate the security incident and at the same time keep everything else up and running.

Another consideration is the cost of people, and time, and resources that will go into the mitigation strategy that you choose. If you're deciding between a number of different strategies, one that might cost \$1,000 and another one that might cost a million dollars,

then you can start to decide which one is the better use of your business resources. You also need to consider how well you're able to contain this particular security incident. If this happens to be something that's spreads very quickly in your organization, then you may have to take some very drastic mitigation steps to contain that, or if this is very slow moving you may be able to move around it and still maintain uptime and availability for all of your other resources. And of course, the decision you make on which mitigation step you go with will also take into account how long it takes to actually put this plan into place. You want to have this contained as quickly as possible.

With technology it's generally a bad idea to let these security incidents play themselves out and run their course, because generally there's a lot more damage that occurs if that happens. And in some cases these incidents have been going on for quite some time already. So you want to very, very quickly mitigate and contain any problems that you might find. You also want to consider how fast this particular problem is moving. This is something the jumps very quickly between systems, then you'll want to get your net around and contain that problem, as quickly as possible.

One way to isolate the bad guys is to put them in an environment that looks exactly like your normal environment, except it really isn't. You've created a virtual world, a sandbox if you will, for them to go inside of and you can watch what they're doing. This way you could start to understand a little bit more about what they're trying to do, because they're attacking a fake network. And you can then use that information to protect your real network.

Some malware is looking for an open linked to the internet. And if you work to disconnect your internet connection then it performs a different set of functions, maybe it deletes itself or it deletes other files on your systems. So you want to be very particular about how you contain these systems and incidents in your network.

**Tags:** certification, comptia, incident, isolation, mitigation, security

**Category:** CompTIA Security+ SY0-401

### **Lessons Learned from Incidents – CompTIA Security+ SY0-401: 2.5**

After an incident has occurred, it's important to compare notes and plan for the next attack. In this video, you'll learn how to answer the tough questions about a security incident.

After a security incident is over it's a good time to sit down and examine what occurred during that incident. None of your systems are going to be perfect. So this might be an opportunity to learn what happened so that next time you can solve the problem even more efficiently. A post incident meeting is a great way to do this. And you should invite everybody who was involved during the incident. That way you'll get the widest perspective, and understand how you can affect change across many different parts of the organization. You should also do this very quickly. The ideas and thoughts that occurred during an incident tend to fade over time. So if you can get everybody into a room very, very quickly after the incident is over it will be fresh in everyone's mind.

One of the obvious questions to answer is what happened? And you should be able to take all of your evidence to March backwards through time to the point when the incident first occurred. You may have to gather information from many different systems, across many different logs, to be able to understand exactly what happened during the incident. Once your incident plan was put into effect how well did it work? You should be able to look back at your plans, and examine were you able to follow the plans, if you did follow the plans, did they work as well as you hoped? Knowing that information you can then determine if you should have done things perhaps a little bit differently and then you can plan to do them differently next time.

Being able to get views from many different people in the room allow you to have an even more detailed plan of attack for next time. And of course, an early warning system is very helpful. So, perhaps, there were indicators that might lead you to this particular incident in a much more rapid fashion. That way you could stop the problem before it ever became an issue in your environment. By analyzing your response of this incident, and getting a complete understanding of everything that occurred from the very beginning to the very end you can start to plan for the next incident. And hopefully, either keep it from your network completely or be able to resolve it much faster.

**Tags:** certification, comptia, incident, lessons, questions, security

**Category:** CompTIA Security+ SY0-401

### **Incident Reporting – CompTIA Security+ SY0-401: 2.5**

What's the best way to gather information during a security incident? In this video, you'll learn some techniques for gathering details and how to report after the security event is over.

When you're in the middle of a security incident there is a lot of data that's being generated. Some of it is in log files. Some of it is messages that appear on a screen. Sometimes you notice physical things with the environment around you. So you need to gather pictures, log information, and data, and as much of it as you possibly can. Whatever you're gathering though needs to be very objective and very factual. You don't want to jump to conclusions. You want to stick directly to the facts and have all of that available in however you're capturing that data. One way to capture that data is with a log book. You can physically write into this book the things that you were observing and you'll be able to preserve it, well, forever. It's on paper and there's not a possibility then that you could lose it due to some type of electronic means.

We're carrying digital cameras around with us all the time now, as part of the mobile devices that we have, so we're very easily able to take pictures of anything we might come across. Even if it's messages on a screen it's great to be able to capture that instead of relying on log files after the fact. I don't have any problem using a log book to document what I'm seeing, my problem is when I try to read what I've already written into that log book. My penmanship is not the best, so it may be more appropriate to use something like an audio recorder. Or I could simply speak into a microphone, document what I've seen, and then later on transcribe that onto a piece of written paper, or type it out. And of course, having a central place to store all of this reporting information would be ideal. So it may be good to have a laptop that is dedicated to incident reporting and it could be something that you never connect to your internal networks.

When you're tracking this information about the incident there's a number of different items you may want to document. One is the status of the incident. You may want to keep an update, as you go through the process, of what was happening at any particular time. It may be useful to have summary information, as well, so you can roll up a lot of larger amounts of data into a small description, that can then be used to communicate with others. It might be useful after the fact also to look at all of the different incidents, and see if there's any relationship or commonality between each one of those. There may be a central point where you can resolve or mitigate the risk associated with these particular incidents.

Any time you make a change or perform any type of action relating to an incident you should always document that. Especially, since you'll want to examine the results of that

particular change after the fact. If you're collecting any evidence about the incident, whether it's physical evidence or digital evidence, you'll want to have a chain of custody. So you know exactly what happened to that evidence, and who may have had access to that particular piece of information. You may need to go back to this incident a number of times, and it may take a number of years to be able to analyze everything associated with the incident. So you want to be sure that everyone's contact information is well documented within the incident reporting materials.

Each individual is going to have a different perspective of exactly what they saw and what they did during the security incident. So it's always useful to have individual written comments from all of those different incident handlers. You may be able to find a piece of information written by one person that didn't show up on anybody else's report. And ultimately, we need to understand what to do next. And after analyzing an entire incident set of reports you may want to create some objectives of things to do next time. So that you can minimize the impact that these security incidents might have.

**Tags:** certification, comptia, incident, reporting, security

**Category:** CompTIA Security+ SY0-401

### **Incident Recovery and Reconstitution – CompTIA Security+ SY0-401: 2.5**

When the security event has concluded, it's time to rebuild any damaged sections of the network. In this video, you'll learn some strategies for getting back up and running as quickly as possible.

Once a security incident has occurred we obviously want to get things back to normal as quickly as possible. We want to get rid of anything that might be bad inside of our environment. Of course, keep everything that might be good. But of course this is much easier said than done.

If you have been infected with malware for instance we want to eradicate that bug. You want perhaps remove the malware, get rid of any user accounts that may have been breached, and fix any vulnerabilities inside of the system that caused that malware to infect the system to begin with. It may not be as easy though as simply deleting the malware from your computer. Modern malware is very good at embedding itself inside of your computer, and you may never really ever be 100% sure that you've eradicated every piece of that malware. In those cases you may want to recover the entire system, perhaps, recover everything from backups that you have so that the system is back to a known good state. You may not have backups in those particular cases, you may be rebuilding the entire system from scratch. In any case, you may want to be sure that you simply replace any files that you may suspect have been compromised. And then of course, you want to tighten down the perimeter so that you can be assured that the malware won't make it self back into your computer later.

In these larger attacks it may not be as simple as recovering a single computer and being back up and running. The reconstitution may be much broader and may involve many, many different systems. So you may require a phased approach to get everything back and running at 100%. Sometimes the phased approach to be over a weekend, sometimes over a month, and in some cases it may take a number of months to finally get back to an original state. These attacks when they occur can be very invasive. They get inside of your environment, and they embed themselves on as many systems as possible. So it may take some time to inventory everything in your environment and understand exactly what systems were affected by this attack.

To make the plan as efficient, as possible, you should consider breaking things up into small pieces. And you can start working on the easy quick things first and leave the much longer implementations for afterwards. So if you can hit those very high value and important resources, and resolve those very quickly, like patching systems and changing

firewalls, you can then concentrate on the more long term fixes, like changing pieces of your infrastructure to make it more secure. Or doing very large scale security rollouts of new firewalls and new intrusion prevention systems. The goal is to make this process as efficient as possible, and if you have to reconstitute a very large part of your network you may want to consider using this particular methodology.

**Tags:** [certification](#), [comptia](#), [incident](#), [reconstitution](#), [recovery](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **First Responder – CompTIA Security+ SY0-401: 2.5**

The first responder to a security incident has a number of important responsibilities. In this video, you'll learn the roles of the first responder to a security event.

Your Incident response policy should have a very detailed section on what do you do when you're the first responder. If you're the person who comes across this problem, comes across this issue, what do you do? And it needs to be well documented, in a lot of detail, because there's many things you can do when you first arrive on the scene.

One of the things that's very important is not to disturb the environment or to only disturb as little as possible. You want to be able to go back later, and recreate what occurred, and find information about what was there. Normally, you would have multiple people involved. You have a phone list, you call some people and you say, what do we do with the system? You want to be careful that you don't damage any evidence that might already be there. You want to then follow the escalation policy for your organization. Again, this is something that's documented.

Who do you call first? If this system is one of our incredibly important systems maybe we also bring in a director, or vice president, or CIO, or even higher level within the organization. There should be a call sheet. You should know immediately who to communicate with, and how to get the people on site that you need to resolve these types of issues, so that you can gather as much information as possible, inform as many people as possible, and have what you need later on to piece together what really happened.

**Tags:** [certification](#), [comptia](#), [first responder](#), [incident](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Data Breaches – CompTIA Security+ SY0-401: 2.5**

Nobody likes their data to leave the network, but sometimes data breaches occur. In this video, you'll learn how to secure and analyze the attack after a data breach has occurred.

It said, that data is the most important asset in any organization. So obviously, we need to protect the data as well as we can. If that data is stolen then we have a bit of a problem. It's certainly too late to recover if it goes outside of our organization. In those particular cases copies are made of the data. And in the case of credit card information those numbers are easily sold on the internet. That data obviously would no longer be confidential.

So now, we have to go into a recovery mode and determine, exactly, what data was indeed taken. You have to identify this, and in some case identify the maximum amount of data that could have possibly been stolen, that may be difficult to determine. But it's important, especially, if we want to identify in the notify users that their data is now stolen. If data has been taken it may be very important to identify, exactly, who took this information. You may have to involve law enforcement, and they may be able to help with the identification process. Identifying this person may also allow you to stop any future breaches from occurring as well.

If you know you've been breached, then it's time to go into recovery mode. So you want to look at every possible device that the bad guy could've touched, and you need to make sure that it is secure. So you want to change passwords and update firewall rules across your entire organization. Even if it doesn't appear that a system has been breached, it's still a good idea to make sure that all of your passwords have been changed. That way you could be assured that nobody has a list of passwords, or could possibly get in there with some old credentials. You also need to notify everyone who might be affected by this breach if there was data that contained private information from customers, or partners, or employees. You'll need to make sure they're aware that this information may have gotten out.

And in the case of modern **HIPPA** and **PCI-DSS** requirements you may be financially obligated to sign each one of those users up for credit monitoring at your expense. So it's very important to understand, exactly, the scope of this data breach and make sure it's a very accurate, so that you can then plan for what you need to do in the future.

**Tags:** certification, comptia, data breach, incident, security

**Category:** CompTIA Security+ SY0-401

## **Incident Damage and Loss Control – CompTIA Security+ SY0-401: 2.5**

Nearly all security incidents will incur some level of damage or loss of data. In this video, you'll learn how to limit the spread of the damage.

One of the challenge we have as a security professional is making sure that when an incident occurs that we're able to minimize the amount of damage, or minimize the amount of loss that has occurred. Somebody was to steal a laptop, maybe we're only concerned about the hardware cost of the laptop, because we have encryption of the entire hard drive on that laptop. So that changes how much damage or how much loss we're really having over that incident.

This does need to be part of your response policy though as what do you do? Is there a way to minimize that? If you walk up to a machine that has a virus on it, it is being compromised by a piece of spyware, maybe we unplug that computer right away. But what if that computer was our primary web server that all of our customers use? Does it make sense to unplug that, if we're relatively certain that virus is not impacting their particular service that they're doing on our website? So maybe we don't want to pull it

from the network, maybe we want to simply partition off, or in some way minimize the impact of that virus to our end users.

That's one of the challenges we have is determining how far do we go. We don't want to cut off our nose to spite our face. But we still want to be sure that our organization is protected and that our systems are secure. Every case is going to be a little bit different. And you as a security professional have to be knowledgeable enough about what you're seeing. And you also, very often, are communicating with others within your organization to make a determination of what you're going to do. Do we turn this computer off? Do we unplug it? Do we capture the hard drive information? Do we put a replacement in place? You now need to make all of those very, very difficult decisions, because they all are going to have an impact later on what you can do with the system and how much information you're gathering. They're also going to impact the uptime and availability of these very important resources for your organization.

**Tags:** certification, comptia, damage, incident, loss control, security

**Category:** CompTIA Security+ SY0-401

### **Security Policy Training and Procedures – CompTIA Security+ SY0-401: 2.6**

Your security policies won't be very useful if your user community isn't trained. In this video, you'll learn some techniques for training your users on your organization's security policies.

You've spent hours and hours putting together your formal security policies. Now it's time to tell everyone all of the policies and make them aware of the things that are important to keep your organization secure. And what better place to put all of this information but in a central repository that everyone can access? And that would probably be your intranet pages. Unfortunately, just putting this information on the intranet pages is not going to make people read it. Fact, they usually will not read the information that's on the intranet, so we have to think of other ways to get this information into the hands of the users.

One good way to do this is with training classes. In the case of the internet and network security, it's probably going to be mandatory training classes. It's best if you can fit this into someone's normal training. Maybe you get some time before or after a normal group meeting that occurs every week or every month. But this does get everybody to see you, to meet you, to understand some of the challenges that your organization has with security, and you get to answer questions from them. Some of the things you would probably talk to your end users about deal with basic security. How to deal with viruses. How to watch for people that are coming in as visitors. You want to make people comfortable with approaching strangers in your building and asking them for a visitor badge.

Or make sure that everybody is aware of the policies. Maybe in your organization there's a policy that everyone must be escorted at all times if there is a visitor, so if anybody sees anyone walking around by themselves, you know something is not quite right. And by empowering your users, they may be more comfortable approaching someone and asking them for their visitor badge or their company employee badge.

You might also want to have specific training for people that have unique security challenges. If someone is getting a new set of laptops, some new tablets, some new mobile devices, maybe you would like to customize some security training around that so that they are really understanding the security challenges specific to those devices. And that way, they'll be more comfortable with the devices, and you'll know that they're trained now to look out for things that are very specific to this new technology.

It would be good if we could customize this training for the specific role that a person might have in the organization. For example, someone in the accounting department probably has different security requirements than somebody in shipping and receiving. But just by looking at a list of everyone's name in the organization, you really can't tell what those differences might be. What is the difference between someone who has a manager role and a vice president role? How does the data access differ for those people? This may take a little bit of research to really determine the type of training specific for those people.

We want to go through and look at all of the different employees, and maybe group them together. Maybe it's not by manager or vice president, but maybe it's by different departments, and the entire department gets a certain type of security training. Every organization is going to be a little bit different in how that role-based training will be rolled out. You'll probably want to have different levels of training, as well. Some people will just be at the beginner level for understanding security challenges. Other people may need more advanced training, especially if they're dealing with very sensitive data or they're using equipment that is very important to keep secure. If you're in the IT department, you're probably going to need a completely different kind of training, since a lot of the requirements for security are very specific in IT. You're in charge of the entire organization's data, so it's important that you get the training that's going to protect not just the IT department, but the entire organization as a whole.

**Tags:** [certification](#), [comptia](#), [procedures](#), [security](#), [training](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Personally Identifiable Information – CompTIA Security+ SY0-401: 2.6**

Our personal information is more at risk than ever. In this video, you'll learn about personally identifiable information and some of the security concerns surrounding our personal privacy.

If you work in technology, you will eventually be faced with the challenges of PII it stands for Personally Identifiable Information. PII might be things like an address, a telephone number, perhaps a picture of someone, maybe a credit card number or social security number.

All this information is very private, and it's something that needs to be handled a little bit different than other types of data. And it needs to be well spelled out in your security policy, not only so that the internal group within your organization understands how to handle the data, but you want to be sure that your customers know how you're handling their personal information.

Maybe this data is something that's stored in your database for a very short period of time, and after you're done using, it you dispose of it. Or maybe it's kept for very long periods of time. This needs to be well spelled out in your privacy policy.

This is something that we almost forget about when you're using it so much in the daily use of your job. This is in a call center, you may be seeing people's identification on your screen and interacting with them all day. But even though this becomes something very common for us to see on our day to day jobs, it's still important to remember that it's personal data and it needs to be handled with sensitivity.

The challenges around PII can clearly be seen when there is a type of security breach of your personal information. Good example of this was in July of 2014, there was an advisory put out by the **Department of Homeland Security National Cyber Security and Communications Integration Center in the United States**. And this advisory

warned that in certain hotels, specifically in the Dallas fort worth area, that there were key loggers that were put on to the computers that are in business centers.

This is very common in hotels, or if you're staying a hotel there will be computers that will have printers, connections to the internet, that will be accessible to you to use when you're staying at that hotel. Well, the bad guys know this, and they stepped into the business center and installed malware that specifically captured keystrokes on those devices.

And so if you went into this computing center, into the business center, and you logged into your email or your corporate VPN, the key strokes that you're typing on that keyboard may have found their way back to the end users. And the advisory said that the suspects obtained large amounts of information, including PII such as bank, retirement, and personal webmail accounts.

This is the type of information we want to be sure in our organization that it stays safe and secure. And you want to be sure that when you're dealing with PII that you have very well established policies so that you know exactly how to handle that data.

**Tags:** [certification](#), [comptia](#), [pii](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Information Classification – CompTIA Security+ SY0-401: 2.6**

Not all data has the same level of classification. In this video, you'll learn about some common data classifications such as unclassified, confidential, and secret.

Not all data has the same sensitivity. There's a big difference between publicly available data that you might find on the internet and personal private information that you might keep close to home. In fact, you could see that internal documents and customer data and inventions that might be used to patent information all may have different sensitivities associated with them.

It's very common then to create classifications that we can then apply to different types of data. So you might seek confidential or top secret, or unclassified, or internal use only. And we can take those labels and apply those labels to types of data that we might have inside of our organization. We can then apply different permissions based on what those classifications might be.

**Unclassified data** might be public data. This might be something that would be available publicly on our website. And there generally aren't any restrictions for anyone to view that particular data. This is different than for instance, classified data, which would be private data, or restricted, or internal use only. This certainly has restricted access and it may require that someone sign a non-disclosure agreement to even gain access to view that data.

**Confidential data** is generally classified on the lower end. But this is still sensitive information and you generally have to have approval to gain access to it. The next step up then would be secret. And that's more of a medium level classification, where you are even restricting the amount of access to that data even further. And the highest level of classification might be considered top secret, where you have to have the highest level of access to view that data.

**Tags:** [certification](#), [classification](#), [classified](#), [comptia](#), [confidential](#), [data](#), [secret](#), [security](#), [top-secret](#), [unclassified](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Data Labeling, Handling, and Disposal – CompTIA Security+ SY0-401: 2.6**

What happens when you need to get rid of data? In this video, you'll find that the handling and disposal of data can be a relatively complex issue.

If you walk around an organization, you'll see **DVD-ROMs**, **CD-ROMs**, even floppy disks that have been thrown into a box, put in a corner, they're stored somewhere for later. But that information that's on those different pieces of media probably has some important company details on it. It certainly could, but you're never quite certain unless you've gone through the extra step of making sure that you label and catalog everything that's on there.

This data tends to stick around for a very, very long time. And if you happen accidentally throw something out and somebody goes through the garbage and notices there's a **CD-ROM**. And they find on that **CD-ROM**, there's a lot of private company information, there could be a problem there. And Unfortunately these things happen a lot.

You see it in the news, every week it seems, that somebody's found some private information that they should have not had access to. So you want to keep track of it and document everything. Make sure that you say this particular DVD-ROM or this information that's on this particular media has this information inside of it. This is confidential information, this is top secret, this is company internal use only. And document those things.

Not just the media you're using, but think of the backups. All of these backups that you're putting together and probably sending off site are also documented, and certainly should be labeled. If there is a set of backups that goes missing, you're going to want to know what was in that backup list. What happened to that cabinet of backups that we sent or that set of disks that we sent off site. You need to understand what the impact of that's going to be.

The disposal of this information really becomes a bit of a legal issue, especially if the data that you have on this media is extremely sensitive. Sometimes your in a organization where you're not able to dispose of information. If you're a government facility, if there's health care, if there's legal requirements that are wrapped around that data, you may have to keep it around for a number of years.

And that means you're going to have to take it off site, you're going to make sure it's labeled. If somebody shows up five years from now and says, where's that information you're supposed to keep? You're going to need to go back five years into your vault and into your storage and pull that information out and say, well, I documented this five years ago, so now I can provide it to you very, very quickly, and very, very easily. This becomes a problem when people start throwing things into the garbage.

It's very easy for people to show up wherever you throw your garbage out outside of your building and rummage through your dumpsters, rummage through your garbage and your trash to try to find information that they can use. It might be security information, it might be privacy information, it might be information about how you do things internally in your organization— that becomes very, very competitive. So you have to be careful about how you dispose of this data. You want to be careful, especially when recycling.

This is something we've all gotten on the bandwagon and say, we're going to recycle all of our loose papers. But you have to keep in mind that this information you are sending off to a recycling organization may have sensitive data on it as well. Make sure your end

users understand that if the data is sensitive, we have to first shred all the information. Then it could be sent off to be recycled.

**Tags:** [certification](#), [comptia](#), [data](#), [disposal](#), [handling](#), [labeling](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Compliance Best Practices and Standards – CompTIA Security+ SY0-401: 2.6**

The data in our organization may fall under some very important compliance regulations. In this video, you'll learn about **SOX**, **HIPAA**, and **GLBA** compliance requirements.

There have been a number of new compliance regulations and security concerns wrapped round compliance in almost every part of every organization it seems. There are certainly compliance issues relating to health care, relating to finance, and relating to any company that is public or wants to be public certainly in the United States at least. If you are not compliant there can be fines and there can be jail time associated with these things. They are certainly not to be taken lightly.

A very common one the United States is **SOX**, the **Sarbanes-Oxley Act**. You'll almost always see it abbreviated as **SOX**. This is the **Public Company Accounting reform and investor Protection Act of 2002**. And it creates some compliance requirements around how an organization deals with the finances, the assets, and how they run their books. This is a big problem for private companies and public companies in being able to maintain accounting reform and make sure that investors are protected in that.

If you're health care you've certainly heard of **HIPAA**, which is the **Health Insurance Portability and Accountability Act**. These are standards for storing customer– in your case– health care information if you're a hospital or an insurance company, how you use that data, even how you transmit that data across the network. There are particular requirements of how you keep that information safe and how people access that data.

There's also the **Gramm-Leach-Bliley Act of 1999**, the **GLBA**. Privacy information becomes a concern. And if you have an insurance policy, if you have information about your car you may have noticed in the mail– at least in the United States– that you have a document that showed up and says we have your private information. And here's how we're using that. A lot of those requirements of notification of come from that GLBA to make sure that all of your private information stays private.

To give you a feel of just how important this is let me tell you what happens if you are not compliant with the health care HIPAA requirements. You could be fined up to \$50,000, or a year in prison, or both. It is a Class 6 felony. If you're doing this under false pretenses the fine goes up. And you stay in prison a lot longer. If your intent is to sell, transfer, or use that information for a commercial advantage, personal gain, or to be malicious \$250,000 and 10 years in prison.

They're also civil fines associated with this, not necessarily criminal. But your organization may have to pay out money, \$100 for each violation, with the total amount not to exceed \$25,000 for all violations of an identical requirement during a calendar year. And again, that's a problem for organizations that are keeping this data. If you violate any of these there's money going right out the door. And in some cases there may be people going to jail.

If your financial compliance is not up to snuff and you've violated the SOX requirements you could be knocked off of your Exchange. There could be a loss of liability insurance that your directors have. This is actually a pretty big deal. There could be multi-million dollar fines. There is imprisonment, a lack of investor confidence certainly, and if you're the **CEO or CFO** that sends this information in and it is wrong, you could be fined up to \$1 million personally. And you could be thrown in jail for 10 years. If you're doing this

willfully– that you're trying to get around or make people think something that isn't really true– \$5 million. And your prison term can go up to 20 years.

So we're talking about some very extensive penalties if you are not compliant with some of these requirements. And as a security professional one of the things you have to always keep in mind is what you're doing internally and how it affects the compliance you have with some of these requirements.

**Tags:** certification, compliance, comptia, data, glba, hipaa, Sarbanes-Oxley, security, sox

**Category:** CompTIA Security+ SY0-401

### **User Habits – CompTIA Security+ SY0-401: 2.6**

How are your users handling the organization's data? In this video, you'll learn some techniques for maintaining good data hygiene for your entire user base.

Sometimes users have some bad habits relating to security and you often have to make them aware of the things that are going on. It's very common for instance, to find at least somebody in your organization who has yellow sticky notes set plot right on their monitor with passwords and other identifiable information on there. So you have to make people aware that that's not something you can really do in your organization.

Also, let people know how to handle the data. Where do you store data on the network? Do you put it in a public folder? Do we have private folders set up? To the end user, those folders may look exactly the same. But of course to other people in the organization, they're different rights and permissions set up on those.

There are many times something called a clean desk policy, which means at the end of the day, or if you leave your desk, everything has to be cleaned off. You can leave nothing on your desk that people would be able to see. Your computer has to be locked and all of your papers have to be put away. And that becomes a habit that people have to get into to be able to do that.

We also have a challenge these days with personal information. You're bringing your mobile phone into the office, you're bringing your tablet computer into the office. And these are third party devices that have the potential to take private information or company information out the door. So there needs to be security policies wrapped around that.

And another way that is very, very common for people to get into environments, more common than you might think, is to simply fill their arms up with boxes of goodies, doughnuts, and other sweets, and ask somebody to open the door for them. You would be surprised how easy it is to get into a building that way. And donuts they're so off putting and everybody wants a donut, it becomes very easy to walk in the door. But you have to train your users that that person, even though the arms are full, they're going to need to badge in, they're going to need to sign in, or do whatever is the standard process for getting into your building. You can't just allow someone in because they've got their arms full.

Make sure that all of these user habits are things that are considered. And that your people get into the habit of doing the right thing when it comes to security.

**Tags:** certification, comptia, data, password, security, tailgating

**Category:** CompTIA Security+ SY0-401

## **New Threats and Security Trends – CompTIA Security+ SY0-401: 2.6**

The security landscape is constantly changing. In this video, you'll learn about some of the latest threats and emerging security concerns.

There are also some bad threats out there your end users need to be aware of. One obvious one is viruses. There are thousands and thousands of new viruses every week. It's very, very difficult for an anti-virus program, a single anti-virus program, to get all of them. Many organizations will put anti-virus at their gateway, they'll put anti-virus on their email servers, and, of course, anti-virus on their end user workstations and their servers. And that's one of the things is, they'll often mix and match different manufacturers of anti-virus in an effort to try to get as many of these viruses filtered out as possible. Obviously, we're going to need new technology for virus someday. We're approaching the maximum capabilities of what we can do with anti-virus. So we're slowly working towards new technologies and new ways to identify some of these viruses.

Another very common threat these days is phishing, where you may be presented with a page that looks just like a login page for our intranet or a login page for Facebook or for Twitter, and we're typing our credentials in so we can log in, but, of course, in phishing it's not really Facebook. It's not really Twitter. It's not really our corporate intranet. We're typing this information directly into the bad guys' web server. And as soon as we type that in, they have our username and password. Yet another threat. We have to make people aware that when you're putting in your username, your password, or any other personal or identifiable information, you need to double-check and make sure that you're going to that site directly. You didn't just click a link in an email to get there.

Spyware is something that thing gets embedded on someone's machine. Maybe they've clicked the link and that link has now put spyware on someone's machine. And now it's capturing keystrokes. It's watching where people browse. It's gathering other information about what may be inside of an organization. Unfortunately, spyware can also cost you money. If you're a financial person that's logging into your bank accounts and moving things around, a key logger can capture all of that login information, send it back to the bad guys, and then from their side they can log into your bank account and do whatever they like with your money. It becomes an unfortunate situation.

There are also exploits called **zero-day exploits**. This is when a piece of software you're running on your computer is vulnerable to a particular kind of attack that up to this point nobody knew about. And now suddenly today the bad guys are taking advantage of that particular exploit. And now the manufacturer of that software has to now come up with a patch. But in the meantime, this exploit is active, and people would be able to take advantage of that on your computer. The only defense of a **zero-day exploit** is very, very quickly reacting to it, and making sure that you've got the patch and you have the information you need to protect that machine immediately. The longer you take, the longer that exploit will be available to the bad guys.

**Tags:** certification, comptia, malware, security, spyware, threats, trends, virus, zero-day

**Category:** CompTIA Security+ SY0-401

### **Social Networking and Peer-to-Peer Security – CompTIA Security+ SY0-401: 2.6**

A single peer-to-peer user in your organization can be a significant security risk. In this video, you'll learn why peer-to-peer software and social networks should be carefully managed.

**Social networking** and **peer-to-peer networking** are technologies that from a security perspective can be crippling to an organization. It can be very, very easy to get your private information out. Good example of this is in February of 2009, the Center for Digital Strategies at Dartmouth College did only two weeks of research. And in that two weeks on a peer-to-peer network, were able to find files, and overall get about 20,000 patient records that had names, and social security numbers, and insurance codes, and other personally identifiable information.

And they had patients that had AIDS. 201 of the patients had mental diagnoses. 326 names, social security numbers of people we're diagnosed with cancer.

This is some very, very private information that should have never been made available to anyone. And there it was something publicly available on a peer-to-peer network, because unfortunately somebody in that organization had installed peer-to-peer software on their computer, not realizing that when you do that you essentially become a file server. And that peer-to-peer software is very good at finding every type of file on your network. And in many cases, making those files available to the world.

All of your content, all of your private information, all of the things that you thought could only be inside of your organization is now available to the world for anybody to access whenever they'd like. We also have challenges with these social networks, because we trust the people that are in our list of friends. And if we get a link from them that says, oh, I saw a picture of you, you should click here to see that picture.

You may not realize that their computer was already compromised. That compromised computer is sending you a link that then is going to compromise your machine, and so on, and so on. So you have to be very careful and make sure users know that participating in peer-to-peer networking puts the entire organization at risk. And just because you trust somebody on your social networking website, doesn't mean you should always trust every link and everything they're going to send to you.

**Tags:** certification, comptia, peer-to-peer, security, social networking, threats

**Category:** CompTIA Security+ SY0-401

### **Gathering Training Metrics – CompTIA Security+ SY0-401: 2.6**

Without some feedback mechanisms, you won't have a way to evaluate your training strategies. In this video, you'll learn some techniques for gathering metrics about your security training.

So how did your security training go? If you're going to be training groups of people, it's useful to know how effective that training was, so that then you'll have an idea of how risky things are going to be in the future. You need some method to assess how well they receive that information. Usually there are checks and measurements you can have in place that can give you some of that feedback.

One method is called a **formative assessment**. You're forming your opinion by constantly monitoring the training throughout the entire process. That way you can make adjustments as you go along, and can really target areas that might need a little bit more work. A much more **high-stakes assessment** is a summative assessment, where you're finally summarising everything at the end of the training. This is usually done through a final exam, or, in the case of Security Plus, through some type of certification exam.

Although we think of somebody sitting in the room and checking things off of the clipboard, most of this assessment process can be completely automated. It's very easy to do this, especially over a large scale. And you want to automate to make sure you can receive as much information as efficiently as possible.

One very common centralized way to do this is through something called a **Learning Management System, or an LMS**. This learning management system allows you to consolidate everything to a central set of systems so that you can go to one place to evaluate and assess how things are going with your training. This LMS gives you very detailed feedback about how your training is going. If you wanted to see everybody who watched a particular video, or you wanted to see who read through a piece of text, or you want to see specific test scores, you can have all of that consolidated within the LMS. This also allows you do tracking over a long period of time across multiple modules, so you can really find every individual and know exactly how their performance is throughout the duration of the training. Not only can you see how your students are doing, the students can also communicate back with you. And having this learning management system in place allows them to leave messages in notes, and you can communicate interactively back and forth all through this central training mechanism.

**Tags:** [certification](#), [comptia](#), [lms](#), [metrics](#), [security](#), [training](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **HVAC, Temperature, and Humidity Controls – CompTIA Security+ SY0-401: 2.7**

The environmental controls in your data center are an important part of your infrastructure. In this video, you'll learn some important considerations regarding your HVAC, temperature settings, and humidity controls.

In a data center, you'll see the abbreviation **HVAC** quite a bit. That stands for **heating, ventilating, and air conditioning**. And if you're in a data center, the ability to keep the environment at a constant temperature becomes a very important thing. All those computer systems they're all generating heat. We have to make sure that they're running optimally. And this is not a simple thing. It's not just you go out and buy some air conditioning and put in your data center. There's a lot of engineering involved in being able to design this properly. It's a very, very complex science. It must be also integrated into the fire system. Not only it this is a very complex system, just to be able to maintain the proper environment, if a fire occurs you don't want your cooling system to circulate oxygen into the environment and make sure the fire is being fed. So that's something that very often is all integrated together and all working together in your environment.

Your data center should be absolutely separate certainly from a heating, ventilating, and air conditioning perspective from the rest of the building. The data center gets very hot if you cooled off the rest of your organization as much as you're cooling off that room everybody would be very, very cold. And if your people are cold, they'll change the temperature and that means your temperature in your data center would rise. So by keeping those separate, you can really maintain the proper amount of cooling and heating. One of the problems in a data center is overheating. So you want to be sure that if something was to happen to the heating system or the cooling system in other parts of the building that your data center would still be in perfect operating condition.

These are systems called the closed-loop recirculating systems and they're also called positive pressurization systems. You'll see those referred to when you look at HVAC. That means in a closed-loop circulating system that the internal air within your organization is constantly being recirculated through there. You're not pulling air in from the outside to be able to do that. And the positive pressurization means that if you open the door to your building air is going to rush out. And that's exactly what you want especially in the case where you have air that is being filtered, you don't want air coming inside that has not been filtered. And in the case of fire or smoke, if somebody opens the door, you would like all that smoke to go out and positive pressurization is going to make sure that everything inside of your building is going to be pushed outside whenever you open the door.

Temperature and humidity in a data center is a bit of a challenge. Your things get too hot, your systems are going to crash. Things get too cold, you're wasting a lot of money on your cooling system. So there has to be somewhere in between that we would be able to use. If you ever walk into a data center, you'll find many of them are very, very cold. But if you look at some of the recommendations from people like Google that has a lot of data centers, they recommend 80 degrees in your cold aisle. We'll talk in a moment what a cold aisle is versus a hot aisle or a warm aisle. Your 80 degree temperature that's kind of warm. But if you look at the manufacturer's specifications for most servers and the components inside of those servers 80 degrees is just fine. That's a temperature that will optimally work for whatever you happen to be putting in. So 80 degrees being kind of warm should be what your data center is. It's remarkable how many data centers though are keeping things at a lower temperature. That's something you'll have to look at whenever you're planning to cool your data center.

Humidity completely different thing. That's how much water is in the air. And when your cooling systems, you're really removing a lot of the water. So you have to also think about controlling humidity in there. If you have too much water in the air, you could corrode your components over, in fact, a very short period of time. If your humidity is too low there's too little water in the air, you're going to have a lot of static discharge and that's something you really don't want around these very, very sensitive electronic components.

**Tags:** [certification](#), [comptia](#), [humidity](#), [hvac](#), [pressure](#), [security](#), [temperature](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Fire Suppression – CompTIA Security+ SY0-401: 2.7**

There are a number of different strategies for fire suppression. In this video, you'll learn about some fire suppression techniques and when they should be considered.

If you're around racks and racks of computers and lots of power, you'll know that water is probably a bad thing. You don't want to bring too much water into the environment. And so in data centers, you generally don't have a lot of fire suppression systems based solely on water. Water is certainly an aspect of it, but generally you're using other methods. And you're finding out if there's a fire based on first detection. Are you able to find smoke? Are

you able to determine if there's flame? There are actually detectors that will look for flames automatically and tell you if it actually sees a fire. And, of course, heat detectors that can identify when certain parts of the organization our certain parts of your data center floor are getting hotter than others.

You might want to be able to suppress with water. It may be a situation where you have a dry pipe, a wet pipe, or even a pre-action pipe of water. A dry pipe means that the pipe is always dry, and you're only going to put water into the pipe if you identify a fire. The negative, of course, is that you have a little bit of time while that water goes through the pipe while things are still on fire. A wet pipe is little bit different. You already have water in the pipe, and there's usually a system that detects heat or detects or melts away, and then the water comes out of the pipe to take care of the fire.

If you are planning to take care of any type of fire with water, there's different methods that you can use. One is a dry pipe, which means the pipe that holds or has your sprinkler system in it is completely dry. And if a fire is identified, the pipe then fills up with water to the proper pressurization and hopefully puts out the fire. The challenge there is that it takes a little bit of time to finally fill that pipe up, but that may give you enough time to determine is the fire a real fire, or is it just a false alarm? You also have the option of a wet pipe, which means you can immediately discharge the water, and it will take care of the fire immediately. There's no delay associated with that. And then there's one kind of in the middle, called a pre-action suppression, where there's already pressure and water in the pipe, but it won't actually turn on until the temperature hits a certain amount, and that causes this pre-action system to go into effect and then start putting the water out onto the fire area.

In a data center, of course, you would like to be able to avoid using water, and we've used chemicals in the past to suppress fires. One that we've used historically is one called halon. It's no longer manufactured. The idea is that you would put this chemical into the air that would reduce the amount of oxygen in the air and would cool things down a bit. Unfortunately, halon also had the side effect of removing ozone from the atmosphere. So we have alternatives that are a lot greener, a lot more environmentally safe, things like Dupont FM-200, for instance. And you'll find that if you're in a data center, there's these big red tanks that are set up in a storeroom, and those will disperse that into the air in the case of a fire.

**Tags:** certification, comptia, fire suppression, security

**Category:** CompTIA Security+ SY0-401

## **EMI Shielding – CompTIA Security+ SY0-401: 2.7**

The shielding of electromagnetic interference an important consideration around computer equipment. In this video, you'll learn about **EMI** and how to maintain proper shielding.

One of the challenges when you get a lot of computers together is they put out a lot of electromagnetic interference. If you've ever had a radio or a telephone near a computer, you may notice that there is some interference coming from the computer. And it comes from the heat sinks, and the circuit boards, and the cables, and the interfaces that are directly on the computer.

You'll find that if you open up a computer, there's a lot of metal shielding. It may be on the case itself, it may be wrapped around different components of the computer. And that's to prevent a lot of this electromagnetic interference. You don't want to remove those because those are preventing those signals from getting into other things that you might have in your environment.

If you ever look at video or if you ever listen to audio that comes right out of a computer over an analog set of headphones for instance, you'll hear the noise. And it may show up in your phone systems, it may show up on your on hold systems. And a lot of that interference is something you have to keep in mind. And keep your computers that are putting a lot of this electromagnetic interference, keep them away from those audio systems.

**Tags:** [certification](#), [comptia](#), [electromagnetic](#), [emi](#), [security](#), [shielding](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Hot and Cold Aisles – CompTIA Security+ SY0-401: 2.7**

A data center cooling system works best when hot air is properly vented. In this video, you'll learn how data centers use these hot and cold aisles to efficiently cool the entire computing environment.

When we talk about hot aisles and cold aisles, we're talking about the way that we're engineering our data centers— where we're putting our servers, and in which rack and what direction on the rack we're putting our servers. If we think about the cooling system in a data center, we've got these **HVAC units**. I put one on both sides of this picture. And this is a side view of all of the racks that you have. So all of your computers, if we were to go down this aisle and look to the left or right, we would see the front or the back of servers that are facing us.

Underneath these data centers are these raised floors, and we have cold air that's going between all these raised floor systems, and it's blowing up into openings that we have in the floor. That is, blowing up into a set of racks where the front of our servers are. We're essentially putting those servers back-to-back, we're putting them front-to-front, if you will, on these servers. And the cold air is moving up, and because it's the front of the server, the server is now pulling that cold air through the system using its fans.

On the other side of the rack is the back of the server where all the hot air is coming out, and so the back-to-back aisles are our hot aisles. That's where all the hot air is coming out. And it's of course raising into the top of the building, where it will then be pulled back down through our air conditioning systems and cooled again and sent through the raised floor.

So when we're designing it to have this as optimal as possible, we'll have certain aisles that will be our cold aisles, where we're pulling all the air through, and then we will have our hot aisles, where all of our air has gone through our computer systems. It's been heated up, and now we're going to send it to the top of the building where we can recirculate it back into and make it cold again.

**Tags:** [certification](#), [cold aisle](#), [comptia](#), [cooling](#), [data center](#), [hot aisle](#), [hvac](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Environmental Monitoring – CompTIA Security+ SY0-401: 2.7**

An effective data center cooling system needs constant monitoring. In this video, you'll learn some techniques for properly monitoring your computing environment.

So we've got all these systems set up. We've created cold aisles and warm aisles. But is it really cooling? Is it really having the effect we want on the temperature? To know, we're going to have to monitor this temperature over a period of time. We want to make sure that what we are cooling is working. We want to make sure that if we raise the temperature a bit to save money that we're not doing this at the cost of the temperature of the systems in our environment. You may not need certain cooling systems. You may be able to turn off different cooling systems. But you'll never know unless you actually monitor these things over time.

So we will want to get a thermometer that can track things over time, maybe provide us with humidity information over time. When night comes you, may have a different pattern than when it is daytime and very hot outside. You may find that different parts of the month are different depending on the number of systems and how much you're working them. Higher CPU utilization will cause more heat in your environment. So we want to log this. We want to look at it. We want to look at those logs afterwards and evaluate, how's our cooling system going? Do we have the proper amount of humidity? Is it working the way we would expect? We want to watch and make sure there are no spikes. We want to make sure there's no outages.

If our system is one where our cooling system fails, you're going to see your temperature rise, so you want your cooling monitoring system to also have alarming alerting capabilities to let you know, if you're walking down the street one night, your phone has a message that pops up on it and says, we just lost a cooling system. The temperature is now risen to a certain amount. Then we can do something about it. We can resolve the issue. We can turn on additional cooling systems, or do whatever we need to do to make sure that we have business continuity, to make sure our systems continue to run.

**Tags:** [certification](#), [comptia](#), [data center](#), [environmental](#), [monitoring](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Physical Security – CompTIA Security+ SY0-401: 2.7**

A proper security strategy includes both technical and physical security methods. In this video, you'll learn about some common physical security techniques.

Let's begin our discussion of physical security with a type of security that we're very familiar with, which is a hardware lock. It can be something like a conventional lock and key that's on a door. It can also be a deadbolt, which is a little bit more of a physical restraint, a much broader lock that you might put onto a door. Many of our doors in our enterprises, though, don't have these keys. They use electronic means to be able to unlock the door.

These are usually associated with some type of central database that only allows the right people into the room based on the card or the identification information they provide to the electronic locking mechanism. Many of these electronic locks are token-based. You need some type of physical token to be able to get into that particular area.

This can be a magnetic swipe with a card reader, or it might have **an RFID** tag inside of it. And you just simply get close and it will unlock the door for you. These are sometimes combined with a biometric reader. So it may take your hand or a fingerprint or look at the retina of your eye. These are multi-factor ways to authenticate or unlock a door.

This might also include something like a smart card and you have to input a pin number and then provide biometrics, and then finally you gain access to the room. A mantrap takes this idea of a locked door to the next level. With a mantrap, there are usually at least two doors involved. And if you open one door, the other doors automatically lock.

If you close that door, the other door can open. This means that only one person can possibly go through at one time. So it's very common for someone to see that all the doors are locked. You authenticate yourself with your card or some other type of access to unlock the first door. You walk inside, and then you have to close the door behind you before the next door will open.

This will ensure that only one person can go through at a time and you won't have multiple people using one person's pass card to get into a facility. Another type of widespread physical security is video surveillance. This is usually done with **CCTVs**, or **closed circuit televisions**. These are cameras that only provide video to internal sources. They're not broadcast to other places. Therefore, it runs on a closed circuit.

And in some cases, you can put a camera in place and you won't need to have somebody physically sitting at that location. It's common to see these outside of doors or entry gates. You push a button and a live person comes on the intercom and asks information about whether you can enter or not. And of course, they can see you, because this closed circuit television camera is right there watching your every move.

If you're planning to purchase some cameras, you need to look at the specifications of what you might be buying. One consideration is the **focal length**. You want to see how wide angle or narrow angle a video stream might be. So you want to be sure to get a camera that fits with the needs that you have. **Depth of field** is another important consideration.

You want to be sure that everything you need to see is going to be in focus. A very narrow depth of field might only focus on a certain area, but a very broad depth of field could view things very closely in focus and very far away in focus as well. You should also consider how much of this camera needs to be able to see even in a dark environment.

You can get cameras these days with infrared capabilities that allow you to effectively see in the dark, although you may need a camera that only needs to be able to record things during the day. So you want to choose the right camera for your needs.

You also might use many cameras throughout the property. You might have them in different locations. And it's very common to network them all back to a central recording unit that also allows you to send signals back out to the camera to be able to control the pan, tilt, and zoom capabilities. Building a fence creates a perimeter around for physical security.

And this is a very obvious form of physical security. And that may not be exactly what you want to do. When you put up a fence, everyone knows there might be something inside that they want. Many fences might be chain link fences like these, which are very transparent. You can see right through them. But of course, you may have a good reason to put up a fence that is more opaque that prevents people from seeing exactly what's going on on the other side of that fence.

Fences make for very good physical borders, because it's difficult to get through the fence, especially if you're using some very heavy steel or you have concrete fences, it would be almost impossible to get through those. And if you build the fence high enough, it's very difficult for somebody to get over that fence. Of course, you could even add on other things like razor wire at the top to help prevent anyone from climbing over or getting through that physical security.

When it gets dark, your physical security becomes even more important. The bad guys love to go places where they cannot be seen. And when it's dark, they can really move around without anybody knowing that they're there. If you were able to use a lot of light, you can prevent this and in some cases, take advantage of the dark by using infrared cameras to be able to see what's going on.

When you're combining lights with cameras, you may also want to think about how you're planning out and designing this light system. You want to be sure that the angles of the lights are important, especially if you're going to be using this video later on to try to recognize someone. If the light is at too sharp of an angle, there may be too many shadows on the face. You may not be able to recognize someone that way.

You also want to make sure that the cameras are positioned so the light is not shining directly into the camera or creating glare. Warning signs can be helpful, especially if your organization deals with chemicals, it's a manufacturing facility, or something like a hospital. These signs will not only help the people that are working in that facility, but will be especially helpful for visitors or people that don't often visit that location.

These signs should consider the personal safety of everyone who's in the building. Your fire exit should be very clearly marked. Or if somebody's coming near a location where there is a lot of chemicals, this would be a great place to have some warning signs. And of course, you should have signs that mark where medical resources might be. If you need some type of first aid kit, it should be clearly marked so that everybody in the building knows where that is.

And even in small organizations, it's good to have contact information or phone numbers right on the sign so people know who to contact in case of emergency. Nothing says physical security more than a physical security guard sitting between you and the inside of the organization. This is truly providing physical protection for the people that are coming into the building and the building resources and people who are already there.

This is going to validate that only the proper people are allowed in, such as existing employees, and the security guard can check and validate that the guest access is provided to the proper people and they are escorted properly throughout the facility. It'll be difficult to look at everybody up in a single list every day. So usually people wear ID badges when they're on site.

And these usually will have a picture and the name of the person and other pertinent details that are important for that organization. These usually must be worn at all times. It's very common to train your users that if anybody is walking around without a badge on, you should start asking them exactly why they're there. Occasionally, you'll have people visit your facility who don't have an ID badge.

And the security guard to be responsible for checking through this list of names and providing access to those individuals. Barricades can also be used to keep people from going into or out of a particular location. There are limits to what a barricade can actually do. But it would certainly make people aware that that is a section they should not be going. This will channel people and anything else through a very particular point.

This might also be able to keep cars or trucks out of a particular area but still allow people to pass through. It's very common to see barriers around industrial equipment, the air conditioning systems, or water systems. And you could avoid having people go near that very dangerous equipment.

You might also see barriers used as another type of physical security. It's not uncommon, for instance, to see concrete barriers that can stop trucks or cars from coming into a building. Or you may even see in very large data centers that they will surround it with water and have a physical moat that separates the data center from a single road in and out of the facility.

If you've ever walked around a home improvement store or any place that's a warehouse type store, you can see the physical network cables being run up in the ceiling. It's wide open.

Well, if your environment is very, very secure, you may not allow your physical cabling to have that level of access. And instead, you may want to have a **Protected Distribution System, or PDS**. With a PDS, all of your tables and all of your fibers run through special conduit that will protect and keep all of the data secure that's inside of those systems.

As it's running throughout the building, it's difficult to secure every inch of every cable. So these PDS's will allow you to provide additional security. This is helpful if somebody's trying to tap into the fiber or the copper connections and be able to gather traffic directly from your network from a place that would not be obvious, if you were trying to do this inside of the data center.

This might also prevent somebody from creating a denial of service condition, where they're physically cutting cables or cutting fibers because there's no way to prevent that without some type of protection around those fiber or cable connections. A hardened protected distribution system would even be one step further, where everything is inside of a metal conduit, everything is sealed. And there are periodic inspections to make sure that everything is exactly the way it should be.

If something's ever out of the ordinary, we want to get an alarm that something has happened. Sometimes these alarms are circuit-based, which means that a circuit is either opened or closed, and then we're notified when that happens. That's something you might usually have on a door or window. And if that door is opened or the window is opened, the circuit closes and a notice is made or an alarm is made at a central location.

It's very useful to have on the perimeter, where you want to find out the instance somebody walks into a door or comes in through a window. You can also have motion detection alarms. They're looking for a radio reflection or even infrared reflections going back into the alarm unit. And then they can then notify a central location.

This is usually put somewhere where you don't expect anyone to be. And if there's any motion, you want to know exactly where that is. A duress alarm is one that's triggered by you. It might be a big red button. And if there's a fire or a panic situation, you can push that alarm and notify a third party. You would commonly use all of these physical security techniques to be sure that the resources and the people in your organization stay safe.

**Tags:** alarms, barricades, certification, comptia, fencing, guards, lighting, lock, mantrap, physical, protected distribution, security, signs, video

**Category:** CompTIA Security+ SY0-401

### **Physical Security Control Types – CompTIA Security+ SY0-401: 2.7**

There are a number of different control categorizations for physical security. In this video, you'll learn about deterrent, preventive, detective, and compensating control types.

When dealing with physical security there are different control types that we can categorize these methods into. One is the **technical control type**. And as technologists, this is one we're certainly familiar with, where we are using systems within our organization to manage this security. It might be controls and rights and permissions that are within operating systems, or it might be things like hardware devices like firewalls and intrusion prevention systems that are all based around technology.

There are, of course, also administrative control types as well. These are policies that help control how people act. So if you're going to set up security policies, formal policies, that you might have in a book or an online resource. Or maybe you have standard

operating procedures, so that people know how to handle a visitor that comes into the organization.

What do you do when somebody is brought on board as a new employee? And what do you do when somebody leaves the organization? Those are all wrapped around the administrative side of those control policies.

When we're talking about physical security, there are a number of different control types that might apply to different kinds of physical security. One is **a deterrent**. A **deterrent** doesn't necessarily keep anyone out of a particular area or prevent access to a particular area. But it does discourage them from going into a room or gaining access to a particular area.

Maybe this is something like a warning sign that tells someone that they probably should not be gaining access. Or this particular area is for authorized personnel only. There may not be a lock on the door, but it may make people think twice before entering a particular area.

Another physical security control is the **preventive control type**. In this case, we are going to prevent somebody gaining access to a room. In this case, we might have a door lock that's always going to be locked. You only gain access to the room if you happen to have the key. Or maybe it's something like a security guard is going to check a list and only going to allow the correct people to enter that particular area.

Another physical control type is the **detective control type**. We are detecting access to a particular area of the organization. This probably is not going to prevent any type of access to the area, but it does give us information about what's going on.

We might have, for instance, a motion detector. And that motion detector's going to cause a camera to turn on and record anything that might be happening in that area. Later on, if we want to do some investigation and find out what happened in that area, we can go to our detective control types to determine what did get detected during that particular time frame. And maybe we'll have motion logs or some actual video footage that we can then compare to those particular time stamps.

And the last physical control type that we will talk about is the **compensating control type**. In this case, we are hedging our bets. We're putting together a plan B so that if something does happen, we have a way to work around that particular problem.

For example, you might be able to have a file server attacked, but we might then restore that file server to a completely different piece of iron using backup tapes. The original server wasn't repaired, we instead worked around and compensated for that by building a completely separate server.

Or you might have a power system back up. If somebody does manage to power down your building, you would simply turn on your generator and your building is back and running again. You didn't repair that original problem, you compensated for that problem by having a completely different physical control type to be able to keep and maintain the availability and uptime of your organization.

**Tags:** certification, compensating, comptia, control type, detective, deterrent, physical, preventive, security

**Category:** CompTIA Security+ SY0-401

## **Business Impact Analysis – CompTIA Security+ SY0-401: 2.8**

When a security event occurs, the organization will need to completely understand the business impact of the event. In this video, you'll learn strategies that you can use to determine the true impact to the business.

**Business continuity** is all about keeping the business going. Making sure that you're able to provide services or products to your end users and your customers. And it really doesn't matter what the incident is. It could be, in fact, pretty far-ranging. You could have power outages or database breach or stolen laptop. But also it might be a much bigger problem. There might be a fire or a tornado or hurricane, something that provides a very, very big challenge to making sure that your organization continues to function if any of these types of things occur.

To get a better handle on what that means, you should start with analyzing what would happen if certain things occur. What are the critical business functions in your environment? You need to understand what your primary business objectives are, and you need to make sure those are documented somewhere, and that you understand what that might be. If you're not able to produce a particular product, you're not able to have people in a building, or if you happen to lose a database, you need to understand how that's going to affect the overall business of what you're doing in your organization. Is it going to provide a loss of revenue? Are there going to be additional legal requirements and people to contact? Is customer service going to suffer if that particular thing occurs? If you lose a database or if you lose a building, it's something to consider as part of your analysis.

You also want to know how long you're going to be impacted. Is this problem going to be something where I'm going to need to bring in additional people? Am I going to need more equipment? Am I going to need to bring in some power generators? Are we going to need to bring in additional resources, third parties to come in? All of these things need to be thought of before the problem occurs so that you can be ready for them. And ultimately, you need to understand— we have us a bottom line, we have either service we're providing or profit that we're trying to make. How's this going to affect us? And if we're going to invest the money in recovering from a disaster, are we really going to see that back in the end and the final bottom line to our organization?

And that's the business decision that has to be made by everybody. Being able to invest in disaster recovery is very often a very expensive thing. It's not trivial to be able to do disaster recovery, to plan, to buy the resources, to test. There's a lot of money involved. And you have to make the decision of that investment that we're going to make in disaster recovery, are we really going to get that back if we happen to use it? And that's something that you have to make a decision very early on, so that you don't go down the road putting all of these **DR**— these **Disaster Recovery**— things in effect. At the end of the day, you may be losing money because of that.

**Tags:** analysis, business impact, certification, comptia, security

**Category:** CompTIA Security+ SY0-401

## **Critical Systems and Components – CompTIA Security+ SY0-401: 2.8**

What are the critical systems in your organization? In this video, you'll learn how to identify tangible and intangible assets and learn strategies for identifying critical business systems.

There are many types of assets within our organization. One of the more obvious ones is probably the people. These are the employees themselves, perhaps the visitors you have coming into your building, or suppliers that you have working with you at your facility. There are also tangible assets. These are assets that we can physically touch. So all of the books that are on a shelf, all of your computer equipment, all of your furniture, anything that might be a paper document, those are the things that we would consider tangible assets.

Intangible assets are the exact opposite. They're assets your company has, but you can't actually touch them. For instance, the branding of your organization is clearly something you can't touch, but it is a very important asset for your organization. Another important consideration are the processes and procedures that you have in place that keep your company going. So things like standard operating procedures, or supply chains are processes and procedures that are extremely important to keeping your business running.

It's important to be able to identify what these critical systems might be. We use these during continuity planning to figure out what we need to protect as part of the plan. So the first step would be to make a list of all your critical systems. This could be a very involved process, especially if you have many different systems and many different processes inside of your organization.

You're then going to want to list out your business processes. These might be things like accounting's that you get paid every two weeks. Maybe it is a manufacturing process that you have in your organization. But all of these very important business processes need to be documented. That's because we're now going to take all of those business processes and try to determine what tangible and intangible assets are associated with each individual business process.

This is an extremely complex process and usually you might even bring in third parties to help you make these determinations, because your accounting system may touch particular servers, there's certain data associated with those, there's third parties you might go to to help with accounting and payroll, there might be a printing process and a printer. And just those individual systems have to be considered if you need to have some type of plan should anything happen to the accounting systems. Although this process is relatively complex and it may take a lot of time to be able to determine what business processes are associated with which assets, it's going to allow you to create a very valuable continuity plan should anything happen to your organization.

**Tags:** business, certification, components, comptia, critical, intangible, security, systems, tangible

**Category:** CompTIA Security+ SY0-401

## **Redundancy and Single Points of Failure – CompTIA Security+ SY0-401: 2.8**

A secure network usually includes a number of redundant systems. In this video, you'll learn more about redundant hardware and systems and how far the planning goes when designing redundancy.

Sometimes problem with business continuity occurs because you have a single point of failure. That obviously is something that if you lose a server, you lose a router, you lose something. That one thing, it can cause everything else to fail. That can really ruin your day, unless, of course, you've made plans for this.

Having that single point of failure can be mitigated. You might have additional hardware you put side by side. Maybe you're making a network configuration of what we call the Noah's ark of networking. You have two routers, you have two firewalls, you have two switches, and they're all redundant. If you lose one, the network will still continue to function because you have a completely different piece of hardware right next to it that's able to take over the load that's going on.

And it's not just networking you need to think about. You need to think about power. You need to think about your facility. You need to think about the cooling system in your data center. If you lost your cooling system, it will not take very long for the temperature to rise and for your computer systems to begin failing.

You also have to think about people and location, especially on things like disasters that deal with nature, hurricanes, for instance, in Florida, something everybody keeps in mind. And if a hurricane comes through, it could decimate an area. There could not be power for days or weeks. You might have buildings that are suddenly here one day and gone the next. How do you handle that?

Do you have people in a different location? Do you get a bus and you ship people somewhere to take over in a remote location for a temporary amount of time? It's something you have to think about because that becomes a single point of failure for what you're doing.

The reality is there's no possible way you can remove every single single point of failure. There's no way to do it. Money is really driving the redundancy. If you had all the money in the world, you could certainly create your own power plants, have completely separate power plants that are providing your particular building or multiple buildings with different power sources. Obviously, not everybody can build their own power plant. So at some point, your single points of failure can only be taken care of or mitigated in so many ways.

And if you keep throwing money at the problem, you can do a pretty good job of that. But ultimately, you have to think about and make a business decision about how far you can go with getting rid of every possible single point of failure. And somewhere in the middle, there's a happy medium that everybody will agree on that will have redundancy in our network, will have redundancy for our facility. But at some point, we're going to have to just rely on that single point coming in and maintain and try to make sure that we can work around that should a problem occur.

Here's a good example of how a redundant network would work. For instance, you have multiple internet connections coming into multiple routers. Those routers are maybe using fault-tolerant firewalls, one that's on standby and one that's always used. And if one of those firewalls loses the magic smoke that's inside of it, it comes out, and it doesn't work any longer, you can fail over to the redundant system.

You might even have redundant core switches in your environment that are going to multiple servers. And even the servers themselves might have multiple network interface

cards inside of them. So that's a very good example, and it's a very common example of how some of the biggest networks in the world are maintaining uptime and availability just by putting redundant systems in in the core and the edges of their networks.

**Tags:** certification, comptia, failure, redundancy, security

**Category:** CompTIA Security+ SY0-401

### **Continuity of Operations – CompTIA Security+ SY0-401: 2.8**

A company's business systems are often interrelated, and the IT department is the glue that holds everything together. In this video, you'll learn about business continuity and some strategies for building a strong continuity plan.

A business and organization is not just the IT department, it's not just the HR department, it's not just the finance department. A lot of these processes that we're doing day-to-day in our organizations involve a lot of different people and a lot of different departments,

The HR for instance, drives the payroll process. IT provides the systems used to process the payroll and maybe even print the checks. And the accounting department of course, provides the money for all of this. So all of those operational functions surrounding things as simple as payroll, are actually behind the scenes are a relatively complex process.

Almost everything in your business is going to rely on IT though. Almost everything we're doing these days goes through our computer systems. So that's probably the first place we'll start is get our systems up and running as quickly as possible. And then blend together all the different software and links we need between all of these different departments.

When you're planning this and your building all of your policies for this, you'll want to make sure you include the entire company with this. Include the HR department, include the payroll system, even the manufacturer of the payroll software, include your accounting department, and everyone else. It is remarkable how interrelated these things are.

And ultimately, we need to make sure we have a document on this. And that's kind of a difficult thing to be able to take all of these very complex business processes and be able to put them in a way that we can understand how they operate, so then therefore we can recover them should a problem occur.

**Tags:** certification, comptia, continuity, operations, security

**Category:** CompTIA Security+ SY0-401

## **Disaster Recovery Planning and Testing – CompTIA Security+ SY0-401: 2.8**

A disaster can be small and large events, and you need to be ready for anything. In this video, you'll learn about disaster recovery planning and some strategies for dealing with disaster events.

When we hear the words disaster recovery, we often think about the big problems– the hurricanes, the fires, the tornadoes. The reality is there could certainly be big problems, but very often there are smaller problems as well. If a water pipe bursts in your facility and that pipe is going to damage systems that you might have in your data center, that becomes a bit of a security problem as well, becomes a bit of a disaster that has to be recovered from.

We will often manage these disaster recovery systems through a third party, or we'll include a third party through this. Maybe we will contract with a data center facility that sits there. And if we ever call a disaster, that data center will be available to us. And we can drive there. It's probably in a geographically diverse location. In case there's something that happens over a large geographical area, we can go to a different city and bring up a disaster recovery data center in that different city.

Generally, we call a disaster. Disaster has occurred. And there's a set of processes and procedures for calling that disaster, because when you start the disaster recovery process, there's some costs that are associated with that. So we're dispatching people.

At that point everything goes into action. We look at our plan of attack. We've gone through all the things that we've been planning for, and now we're actually executing on it. We have to be able to think on our feet, because when a disaster occurs, things might happen that we would have never thought of. We may have realized that we're going to need to have generators if we lose power, but we may not have realized that the power outage would be so massive that we would not be able to get gasoline for the generators because the gas stations don't have power to be able to pump the gasoline.

So all of these things work together, and sometimes you have to think about how am I going to get gasoline. How am I going to be able to pump some of these things out? Sometimes those unknown things can really bite you. And you have to be able to move and change as you go, especially in the middle of a disaster.

If you're going to recover from a situation, you're going to want to test prior to that. You don't want your first time going through to be the incident. And so you're going to plan, and you're going to test for this. Unless you try it, you'll never know what you missed or things you need to add or modify on the plan.

Many people will schedule these tests. They'll do them once a month. They'll do them once a year, and they'll go through the process of understanding how do our processes work. Are they as effective and efficient as we expect them to be or that we need them to be?

So we'll create a scenario. Let's say we lose the building. Let's say this server goes down. Let's say a database crashes. What do you do?

And they go through the process of grabbing the backups, loading them on some new hardware, obtaining new hardware, maybe finding a new building with a new internet connection, maybe failing over to our redundant facility. We want to include as much of the organization as possible during these tests, but we don't want to affect any of the currently production systems going on. You obviously would not fail a real server that's being used on your production network to provide your services to your end users.

You also want to think about documenting this, especially during the testing phase. Should an actual emergency occur and cause a downtime to your business, that's not the time to go through and think if you're doing this right and maybe we should change this for next time. You want to do that during your planning phase and during your testing phase.

And afterwards you'll be able to look at that list and say, what worked during our test. Or what did not work during our test? Do we need to change our processes? Do we need to buy additional resources, or do we need to think differently about how we're going to handle this problem should it occur?

**Tags:** certification, comptia, disaster recovery, planning, security, testing

**Category:** CompTIA Security+ SY0-401

### **IT Contingency Planning – CompTIA Security+ SY0-401: 2.8**

There are a number of formal processes that you can follow for IT contingency planning. In this video, you'll learn about some well-documented contingency strategies from the United States National Institute of Standards and Technology.

There are some guidelines available for IT contingency planning. This is in this publication, **Special Publication 800-34**. And it's called "**The Contingency Planning Guide for Federal Information Systems**." It's something put out by the National Institute of Standards and Technology that is part of the US Department of Commerce. And this has instructions, recommendations, considerations for your IT departments to be able to do contingency planning.

So if you're looking for a guideline, you're looking for something that can help you understand what you should be putting together, grab this document. Google this document and be able to understand what do I need to think about for a relocation and additional systems and how I can do things manually. Maybe we're not going to use our computers and our networks. Maybe we're going to write everything on

Paper. Remember paper? We used to use. We would have pencils are the things that we would use with those.

This document includes three different types of systems that you should consider when you're planning for IT contingency. Talks about client server systems. Those are the traditional systems that we have in our data centers. What about our phones? We talk to people with voice, and our telecommunications systems are an important part of that.

There's also mainframes to keep in mind. And very, very large organizations, you have these monster systems, these mainframe computers. And they become a completely different challenge when it comes to maintaining the IT contingency planning for this, and so this document is going to give you ideas and guidelines you can use to maintain the availability of all three of these different platform types.

This document also provides a guideline of seven steps you should keep in mind when creating or planning for a contingency. The first step is to create a planning policy statement. Come up with what you're planning to do and how your systems will be made contingent, and this will be a formal policy that everybody gets to participate in.

Then you need to look at the business in the overall organization and find out what is the business impact for some of these problems that might occur, and we probably want to prioritize the mission critical systems that we have in our organization. Then we can look at ways that perhaps we can prevent some of these problems from occurring in the first place. That way we can maybe take a big section of disasters or problems that might

occur and be able to mitigate those, be able to make sure that maybe they'll never happen in our organization.

Maybe we're buying redundant systems. Maybe we're creating a different way of backing up our existing systems. And we need to think about contingency strategies, then. If a problem does occur, how can we recover as quickly as possible? And that, of course, means that we're going to create a contingency plan around these things. If a system is to fail and we're going to get up and running quickly, we need to have the exact process and procedures to restore that particular system.

And then we're going to talk about how do we train? How do we plan? What do we do to make sure that these things that we've created are actually going to work when we need them? So there will be testing, testing, and even more testing after that.

And, of course, this is a living document. This is something we'll be improving on. We'll be changing as our IT systems change, this particular plan will need to change. So this will be something that is always going to be a work in progress.

**Tags:** [certification](#), [comptia](#), [contingency](#), [planning](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Succession Planning – CompTIA Security+ SY0-401: 2.8**

Part of a disaster recovery plan is the organization's strategy for succession planning. In this video, you'll learn the important parts of a company's formal succession plan.

Another thing to consider for business continuity is planning for succession. And this becomes an issue when you have people that are leading the company that in many cases are known to everybody within the company, who leave the company, who pass away. These are challenges that may leave a vacuum in leadership or may have a financial impact to the organization.

You never know what can happen. And when it does happen, it can be very, very sudden. If someone is dismissed, if somebody leaves the organization. They were here yesterday and now they're not here anymore, how do we handle that and maintain that our company and our organization continues to function? Sometimes there's a deputy in place.

And a good idea for anybody in a management role is to have somebody who could step in for them. Ideally as a **manager**, as a **director**, as **VP**, your training your replacement. So you should already perhaps even have an office; a deputy CEO or a deputy CIO, that is a formal office of the organization. Should the CIO, the CEO step aside, you'll have somebody to take their place in relatively short order.

There may also be travel restrictions. And this may be part of your entire policy for succession is that if there are people that are leading our company that are traveling, maybe we'd make it so that only one of those people can be on a single flight, or maybe only two board members on a single flight. And that way if something is to happen to an organization, it would have a limited effect on the total number of people involved.

**Tags:** [certification](#), [comptia](#), [plan](#), [security](#), [succession](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Tabletop Exercises – CompTIA Security+ SY0-401: 2.8**

Instead of performing a full-blow disaster drill, you can validate your plans using a tabletop exercise. In this video, you'll learn about tabletop exercises and some techniques for running tabletop exercises in your own organization.

If you really want to see how well your disaster recovery plan is, you would run a test. You would simulate a disaster and then recover from that simulated disaster, and see how you did. But of course, this means that a lot of resources might be involved, and perhaps even a lot of people. And it's certainly going to take time. And certainly that involves money as well to be able to run this simulated test.

Instead, you could do something like run a tabletop exercise. You can determine where your shortcomings might be by simply sitting down and analyzing what you might have done should a real disaster have occurred. This prevents you from having to go through the physical steps of a disaster or physical steps of a drill, but you're still thinking through the process and determining if you have the right plan in place to recover from a disaster.

You want to get everybody together, all the key players, and be able to run through this simulation. This will be everyone around a table at the same time or on the phone discussing what has happened with the disaster and what the next steps might be. And everybody walks through exactly the way it should go based on the plans that you've previously made.

Before you begin doing a tabletop exercise, you need to determine how complex this is going to be. Do we contact the local fire and police departments, or is this something that we're just going to talk about internally within our own organization? Now we determine the scope of this particular disaster. Is it something like a simple water main break that we have to recover from? Or was this a very bad disaster that involves deaths and injuries? And how do we adjust to have those particular problems occur?

Now we need to determine what the scope of the disaster might be. Do we want to have a simple water main break or should this be a hurricane or some type of natural disaster where there's going to be injuries involved? This is going to make a big determination of how far we go through the steps of our disaster recovery process. We want to involve as many people as possible. But you may even want to bring them into the room as a surprise, sit them down, and say, a disaster has occurred. Now what do we do? If you really want to test your disaster recovery skills, this may be the type of disaster that doesn't really give you a warning.

You should also not assume that every piece of information is going to be available. When disasters occur, there are gaps in communication, and it's difficult to know exactly what might be going on. But people still need to be able to make decisions on how to recover from that. So this tabletop exercise should not be a perfect scenario. But you should run through something realistic enough that later on, you can look back at how you did and determine where you need to make changes with your disaster recovery plans.

**Tags:** certification, comptia, exercise, security, tabletop

**Category:** CompTIA Security+ SY0-401

## **Redundancy, Fault Tolerance, and High Availability – CompTIA Security+ SY0-401: 2.8**

If you're planning to maintain uptime and availability of your computing resources, then you'll almost certainly need to implement redundant systems. In this video, you'll learn about redundancy, fault tolerant systems, and high availability infrastructures.

Whenever we think of keeping all of our systems up and running in an environment, we very often think about what can happen if we lose a server, if we lose a router, if we lose another component within our devices. So we have to think about redundancy and fault tolerance. These are very similar ideas, redundancy and fault tolerance. The idea is to keep things up and running and maintain uptime. We want to be sure that all of the systems, all of the things on our network– that we're able to use all of the resources available to us and our company continues to function the way it should.

So we need to make sure, for instance, that we don't have a hardware failure. We may want to have redundant servers. Or within a single server, we may want to have redundant power supplies. And so by keeping those redundancies of those systems, if we happen to lose a power supply or we happen to lose a motherboard in a server, we've got another one sitting right there, ready to take its place so that we can keep things up and running.

We also need to think about the software that we're running on these systems. We may want to get software that's able to notify us whenever there's a problem, or work in conjunction with other pieces of software that might be running, perhaps in a cluster, so that if one particular piece of software fails, you've got other pieces of software running on the same network that are able to pick up the slack should that problem occur.

And we also want to be sure we don't have any major system problems. Maybe we would like to have redundant routers. Maybe we'd like redundant firewalls. Maybe we would like redundant wide-area network links to the internet. You can obviously really apply different types of redundancy and fault tolerance to many environments. So by having these extra systems in place, we can always be assured that our systems will be available and up and running 100% of the time.

Now just because you have multiple servers or multiple systems– you've got that redundancy– doesn't necessarily mean that your environment is highly available. High availability means that the systems will always be available regardless of what happens. With redundancy, you may have to flip a switch to move from one server to the other, or you may have to power up a new system to be able to have that system available. High availability is generally considered to be always on, always available.

If you have multiple high availability systems and you lose one, it doesn't matter. Everybody continues to run because you've got an extra system ready to take up the extra slack, the extra load associated with that resource. There may be many different components working together to have this happen.

You may have extra and multiple wide-area network connections with multiple routers, with multiple firewalls, with multiple switches going to multiple servers, and they're all working together and in conjunction. Each one of those sections would be set up the high have high availability so that if any particular one of those failed, all of the other components can work together to keep the resources up and running in your organization.

Now redundancy and fault tolerance means that we're going to need to have redundant hardware components. So you can already think about having multiple power supplies, maybe having multiple devices available for us to use. We might also want to have multiple disks. Within a single server, in fact, you can have something called **RAID**, which is a **Redundant Array of Independent Disks**. And this **RAID** methodology means that if we lose one disk, we have options to keep the system up and running without anybody ever knowing that there was a problem with that piece of hardware.

Another piece of hardware we may want to have— because we're never quite certain how power is going to be in our environment— is something called an uninterruptible power supply. You'll hear this referred to as a UPS. If we ever lose power, these UPS systems have inside of them the batteries and some other method to keep things up and running.

And those UPS systems can be extremely valuable, especially if you're in an environment where power is always a little sketchy. You may be in the southern United States during the summer where there are a lot of thunderstorms. Power goes on and off all the time. You almost require a UPS on your system to make sure things are available to you.

If you want to be sure that resources running on a server are available, you may want to consider clustering a number of servers together. That way if you lose a motherboard, if a system becomes unplugged. If you have a system piece of software in a system fail, you can have these extra systems in your cluster to keep everything up and running. And since all of those cluster machines are all talking to each other, they know if there's an outage and they'll be able to take those resources and make sure that everybody is able to run all of the systems that they need to run.

You often see systems very, very often load balancing these things. It's very important. If you have multiple systems in place, you want to have all of them running all the time so that you're balancing the load between them. And if you lose one, everybody will flip over to the other. Because the load is being balanced, you'll want to make sure that you have additional resources available on that original machine so that it's able to keep up with the load. It's a lot like having multiple engines on a plane. If you lose one engine, you know that extra engine on the plane is designed to be able to keep that plane in the air until you're able to get it down on the ground safely.

I mentioned that **Redundant Array of Independent Disks** that you might have inside of a single server. There's different types of RAID out there. This chart shows you an idea of the primary kinds that you'll run into. RAID 0 for instance, is a method called striping without parity. What that means is you have multiple disks, and parts of the files are copied to those multiple disks, but only part of the file, which means that we're able to have very high performance because we're writing tiny pieces to many different files at the same time. The problem is, there's no parity, which means if we lose any one of those disks, the entire system is unavailable to us. So there's no fault tolerance associated with that at all.

Another RAID type is **RAID 1**, or mirroring, where we are exactly duplicating this information across multiple disks. So if I have a 2 terabyte disk, I'll have a duplicate 2 terabyte disk that has exactly the same information on it. If I lose the first disk, it continues to run, because now we're fault tolerant. I can use the exact copy of that disk in **RAID 1**.

**RAID 5** is very similar to **RAID 0**. It is striping, but it includes an extra drive for parity data, which means I'm not getting an exact duplicate of the data, but if I lose any of those drives, I still have a way to fault tolerantly retrieve all of that data from the disks. This is a pretty advanced system to be able to do something like that, but it means that if I lose any physical drive, I'm still up and running. And I'm not using the exact duplicate amount of data that I have in RAID 1. So we've got some efficiencies there in the amount of storage in our systems.

Occasionally, you'll see these RAID systems combined with other systems. You might have striping without parity, but you'll mirror that striping. Or you'll mirror the data and have it striped to a parity disk or striped to a non-parity system. So you've got different options where you can combine these things together. So you often see RAID 0 plus 1 or RAID 5 plus 1, where you are doing striping with parity and mirroring all at the same time. A lot of flexibility there, and if you're building these file systems in your servers, you'll want to check and see what RAID options might be available for you.

I mentioned server clustering. That's a really useful way to keep systems up and running, and to provide availability 100% of the time. In an active/active server cluster, all of your end users are out here accessing different servers in your environment. And these servers are always active with each other. They're constantly communicating between each other so that the two systems know if they're available and running. And then you have behind-the-scenes storage that both of these systems will share. The idea is that if you lose one section of this cluster, everybody can still go right to the other active side of the cluster to be able to use those resources.

An **active/passive cluster** is a little bit different. In active/passive, you have one system that is always active and one system that is always passive. The passive system is sitting there and doing nothing. It is waiting for a problem to occur. These clusters are always talking to each other and making sure they're up and running. And if Node 2 notices that Node 1 has disappeared, that the active system is no longer there, it automatically makes itself available to the world. And now all of the clients begin using the backup or the passive system to be able to perform whatever function they need across this network.

**Active/passive systems** are generally much easier to implement, because these are exactly the same type of systems. Active/active tends to be a little bit more complex to implement, because you now have multiple systems talking to multiple servers simultaneously. There has to be a way to keep track of that and make sure that everybody's talking to the right systems at one time. But whether you're using active/active or active/passive, you have systems that are redundant and available should there be any problems on your network.

If you're planning to have redundant systems, you may not have them all running the same way. You may have cold spares, which means you've bought an additional server, but you're keeping it in a box in a storeroom somewhere. You may have 10 servers sitting in the rack, and if any of those 10 servers fails, you can go to the storeroom, pull your one spare out of there— your cold spare— put that in the rack. And then of course, you have to configure it, because this is a fresh configuration.

You may want to have something called a **warm spare**, which means that spare is something that you might have even put into the rack. You'll occasionally have it turned on. You may have it updated with the latest software, updated with your configurations. That way if you do have a problem, you simply flip a switch, turn it on, or plug it in. And now that warm spare is ready to go. You don't have to now perform any additional configurations or load any additional software to get that running.

And obviously, your last option is a **hot spare**. It's always on. It's always updated. In many cases, it's designed to automatically take over should there be a problem. So if you do have a problem with the system, it goes down, you can immediately move to the hot spare and it has an exact duplicate, an exact updated system that everybody can now use to perform the function that they need on your network.

**Tags:** active, certification, comptia, tolerance, high availability, passive, redundancy, security

**Category:** CompTIA Security+ SY0-401

## **Cold Site, Hot Site, and Warm Site – CompTIA Security+ SY0-401: 2.8**

If you're managing a disaster recovery plan, then you'll need some off-site options. In this video, you'll learn about the advantages and disadvantages of cold sites, hot sites, and warm sites.

If you're planning for a disaster, then you're probably going to want to have an off-site facility where you'll be able to take all of your systems and get your organization back up and running should a problem ever occur. So as you're pricing out and looking for different environments, you've also got a lot of different options on how you configure and set up these remote locations should a disaster occur.

If your disaster recovery location is a cold site, well, that means that it is essentially an empty building. There is probably some cooling systems. There is probably other facilities available. But there's really no hardware there waiting for you. There's nothing available for rack space. These are things that you would have to bring yourself should an emergency occur.

And of course your data is not there, either. You need to make sure that you have a way to get to your data backups and that you have a way to transport them to this cold site. You also have no people here generally. So this is in a remote location. You have to think about how you will get people from one facility to this backup facility, this disaster recovery facility so that you are able to keep your systems up and running.

A **warm site** is one step up from a cold site. So it may be a location where you have all of your equipment, but the hardware is stored in a separate room. Maybe it's not out on the data center floor. If a problem occurs and you call a disaster, someone will show up at that facility, start pulling things out of your closet and putting them in racks in your location.

That means all you have to worry about, then, is your data. Bring your backups with you. Bring any other data that you might need. And now you can load it onto these systems that you might have at this warm site.

A **hot site** is the far extreme, where you have an exact duplication of everything. All of those systems are up and running. You've got a complete duplicate of a data center at a remote site. So whenever you're buying hardware, you naturally buy duplicates of that hardware, put it on your hot site, get it up and running. In fact, you're usually updating all of your software, all of your configurations, everything in that site so that you can flip a switch and have everything move from one site to the other. Maybe it's not quite as simple as flipping a single switch, but if you've got a hot site available, it becomes relatively simple to move all of your technical resources, all of your IT infrastructure really from one data center to another in a very, very short period of time.

**Tags:** certification, cold site, comptia, disaster, hot site, recovery, redundancy, security, warm site

**Category:** CompTIA Security+ SY0-401

## **Confidentiality, Integrity, Availability, and Safety – CompTIA Security+ SY0-401: 2.9**

The **AIC triad** is an important concept in security. In this video, you'll learn about confidentiality, integrity, availability, and safety.

The fundamentals of security are often rolled up into a set of principles called the **AIC triad**. This stands for **availability**, **integrity**, and **confidentiality**. The availability part of the triad is referring to systems being up and running. You want to maintain availability of all of your servers and all of your networks and make them available for everyone. The

integrity side means that as traffic is traveling from one side to another, you want to be sure that nobody makes any changes to that information. When it's received, you want to be sure the integrity of the data is maintained all the way through the system.

And with confidentiality, we want to be sure that the only people who are able to view this information are the ones that have the rights and permissions to do so. With confidentiality, only certain people shouldn't have access to certain types of information. We can manage this in a number of different ways.

One very common way is through encryption. You can encrypt information, send it to another. And that person can then decrypt the data, but anywhere along the way you have that data private. Nobody's able to see the information that you were sending.

You can also provide confidentiality through access controls. You set rights and permissions to a file or a resource, and you can apply those permissions to groups of people or individuals so that only those people would be able to view that information. You can even provide confidentiality in unexpected ways, like using something like steganography. This means that you're concealing information and data within another piece of information.

We commonly see steganography used to hide data or information within pictures and then send those pictures across the network or post them to a web page. For people who are surfing the net, they're viewing the page and looking at normal images. But if you're somebody who knows that that information is hidden in the image, you can download it and extract that information directly from inside of those pictures.

In the security world, integrity means that when we send information from one point to another, that information is not changed anywhere in between. And everything that we have received is being received and stored exactly the way it was intended when it was set. That means if any part of this data has changed anywhere in that transmission, that we are aware that this change has occurred.

One way to maintain integrity is to create a hash of what we've sent. And on the other end, after this information has been received, the other end can perform exactly the same hashing algorithm and then compare the original hash with the ultimate hash the was received. This way we're able to be sure that what we received was exactly what was sent.

A more advanced form of integrity might be something like a digital signature. This is a mathematical scheme that allows the sender of the data to digitally sign the information that's being sent. And on the other end, that signature can then be checked. And the signature is also maintaining the integrity of the data. If the digital signature doesn't match when it gets to the other side, then something has either changed with the signature or the data. And clearly there's a problem with the integrity of the data that was received.

The digital signatures usually work in conjunction with certificates. These certificates are used to sign this data originally so that on the other side the certificate is then compared. Generally, certificates are also associated with individuals or resources so you can be sure that the data came from exactly who you expected.

If someone has digitally signed some information and they've sent it to you and you were able to verify the digital signature and the integrity of the data, that's something that we call non-repudiation. That means the person who sent the data would not be able to say that anything had been changed within that. They would not be able to repudiate what was received by you, because you are able to confirm that the information you've received is exactly the same information that was sent.

The idea of availability means that your information is always going to be something you can access. If you need to get a report from a server, it should always be there. If there's a video you need to watch, that video needs to be instantly available.

One way to provide this availability is through redundancy. That means we have multiple systems available to provide access to these services. We might have multiple routers or multiple switches or even multiple servers located in different locations. That way, if anything was to happen, we would be assured that this service would maintain its availability because you'd have a complete duplicate still running somewhere else.

This is very similar to a design that might be fault tolerant. That means there is absolutely a failure of some kind within the system, but it's going to continue to run. In a fault tolerance system, you could even have the system running not as effectively as it was before. But at least the services would still be available.

We don't usually think of patching our operating systems or our applications as availability. But indeed this does help, because you're creating a more stable environment. And in the case of security patches, you're making sure that the bad guys aren't able to affect the availability of those systems.

Another important security concern is the safety of the people within your organization and the data that your organization has as an asset. These are things where you would create escape plans and routes. So if there was a problem with the building or a fire, everyone would know the best way to get out of the building or the best way to get out of the entire area.

To do this, you would commonly run drills to make sure that everybody could get out of the building, go to the correct location. And you could do it as quickly as possible. Once those drills are complete, you can analyze how quickly people were able to get to their proper locations and then adjust and make any changes that might be appropriate.

It's also very common to run digital tests against your systems and your protections to make sure that people don't have access to your data. You want to keep your data just as safe as you keep your people. And that way, you'll be able to maintain the uptime and availability of all of your systems.

**Tags:** AIC

triad, availability, certification, comptia, confidentiality, integrity, safety, security

**Category:** CompTIA Security+ SY0-401

### **Malware Overview – CompTIA Security+ SY0-401: 3.1**

Malware is one of the most prevalent forms of malicious software attacks. In this video, you'll learn about malware types and some of the problems that occur when malware is installed onto our computers.

Malware has become an enormous problem, primarily for Windows-based computers. Windows machines obviously have the vast majority of computers that are out there in the world. So virus and malware attackers have gone after that particular operating system as one that they can make the most with the number of systems that happen to be out there.

Very often, this malware can be very, very bad software. Once malware gets onto a system, it becomes very, very difficult to remove it from that computer. Even worse, it becomes even more difficult to know if you really removed it from that computer. Very often, you can think that you've absolutely removed every piece of malware from a computer, and in fact, there happens to be some left over that then re-infects the machine again.

So this isn't something where you can click a button, flip a switch, and absolutely know 100% certain that you've removed it. Very often, you're never quite certain. Never quite 100% sure that you've gotten rid of all the malware.

Some of this malware can be very, very bad. It can gather information. Once it's on your computer, it can watch every key you press. It can watch you as you're logging into your bank account, see what your username is and what your password is, and send that information back to a central mother ship that then gathers those details to go after your bank account. This is something that happens all the time. There are certain types of malware that are specifically written to take money from your bank account. That's one of the ways that they do it.

These also can participate together in a large group. Botnets, for instance, can be installed onto your computer, and that botnet is simply sitting there and waiting for commands. And when a central repository, a central controller out on the internet decides that your computer should send spam, or your computer should participate in a denial of service against another machine, it comes to life and starts doing that. They're using your computer as a jumping off point to do anything they'd like to do, just because that piece of malware is installed on your computer.

Another big way that these guys make money is through advertising. And if you've ever opened up your browser on your computer and suddenly a bunch of pop-ups show up on your browser, you haven't even surfed to a website yet, it's very possible you may be infected with a type of malware that's adware, that shows you advertising. Because the more advertising they can get in front of your eyeballs, the more money, ultimately, they're going to be able to make. And obviously, viruses and worms, and other types of malware that get on your computer, can damage files, can remove files, can create your computer so that it can't even boot up and perform normal functionality.

**Malware** is an enormous issue, not just for home users, but also for businesses throughout the world. And it's something that we're going to continue to have to fight against to be able to keep it off of our computer systems. There are many types of malware. I want to list a few of them that you'll need to know for this Security+ certification. But there are even more types of malware beyond what we're going to discuss in this.

One very obvious type that we've already discussed is adware, where multiple ads are presented to you. Another type of malware can just be the traditional computer virus,

where now a computer virus is on your computer and creating problems. There's many different types of viruses out there and we'll talk more about those in a future video.

There are also worms that don't need you to be able to move from machine to machine. It uses the internet. It uses the network to hop from one computer to another. A **worm** is a type of malware that can get on your computer and then infect every other computer in your organization, just because it started on your PC.

**Spyware** is another type that keeps track of what you're doing and reports back what websites you've visited, what keys you've pressed, what things you've clicked on with your mouse, and then when said information is sent back, they're able to do things with that like present other types of ads to you or go right into your bank account and use your username and password that you typed in to get to your information.

**Trojan horse** is a type of malware that presents itself as one thing, but in reality, it's really a piece of malware behind the scenes. This is a good example of one. This is a new type of malware called ransomware. It's really a type of Trojan horse that gets on your computer and it says, oh, this is XP anti-spyware. How nice that we have this anti-spyware to take care of the spyware on our computer. And boy, look at all the spyware on our computer that it's found.

In reality, none of this spyware is on our computer. This anti-spyware program has simply identified things, presented things to the screen that aren't there, so that you'll buy this fake software. So it's something that becomes a very big problem. There are other types of Trojan horses to watch out for as well, but that's one that's becoming more and more common out there.

There's also rootkits. These are very, very bad. Fortunately very, very uncommon to find these, because rootkits are malware they can hide themselves from your anti-spyware, from your antivirus, and from the normal types of checks that you would do on your computer. You would have to write a very specific rootkit remover program just to get rid of rootkits.

Back doors are things that are turned on when a piece of malware gets on your computer so that other programs can access the computer externally. First guy in unlocks all the doors and the windows so that everybody else can come in. And that's exactly what a back door would do for you.

Logic bombs a little bit different. They're designed to wait for certain date, a certain time, or certain thing to happen, and then it goes into effect. We'll talk about logic bombs also in a future video.

And lastly, botnets, where multiple computers, in some cases millions of computers can work together and can be controlled from a central place, and that central repository can make your computers do a lot of different kinds of things. We have a different video for each one of these malware types. You'll be able to study each one of these and learn a little bit more about what makes every single one of these operate.

If you've ever gotten that phone call and somebody says, I've got malware on my computer, what do I do. And one of your questions is, where did you get this from? What did you click on? Was this in an email link that you clicked? Is it on a website and something you clicked? How did this get here? And there's many, many different attack vectors that can be used to present or get malware onto a computer. A few of those may be all working together as well. It doesn't just have to be one thing. There can be a piece of software that finds a vulnerability in your computer and is able to take advantage of it.

I had a computer that had an old version of what used to be Sun Java, now it's Oracle Java, running on the computer. It was months old and there was a known vulnerability that this piece of software took advantage of and got on my computer, embedded some

spyware on my computer, and then opened a back door. So if there was another machine that was already waiting to take advantage of this, they could then get in through the back door of my computer and install much more. Fortunately, I believe I stopped it before that happened. But you can see how all of these things work together to get the malware onto your computer.

Maybe it's not a Trojan taking advantage of a vulnerability. Maybe it's you clicking on something that's in an email message that was sent to you that looked legitimate. But in reality, it was installing malware onto your computer. Or perhaps it was a link on a website that took you somewhere to install that malware on your computer. The idea is that the computer you're using has to run this to initially get it onto your computer. So it could be that email link, a web page pop up.

There's also something called a drive by download, where you can have simply a file that starts downloading itself when you visit a website, and it gets embedded on your computer without you telling it that it really, really needs to go there. Drive by downloads can be very, very difficult to stop, because they're taking advantage of known vulnerabilities.

And also worms, you could just have viruses bouncing and worms bouncing all over the network. And worms are viruses and malware that are able to propagate themselves. They don't need you to do anything. They're taking advantage of known vulnerabilities, generally in an operating system or an application. And that's where these malware authors are really writing this code for, are those known vulnerabilities. You need to make sure that your operating systems are always updated, and also the application you use.

I mentioned that one of the pieces of malware I recently saw came through Sun's Java, an old version of Sun Java that I needed to upgrade that was very, very old. They knew that that application, if I was running it on my machine, that it could infect my system. And sure enough, that's exactly what it did.

You may see very often that Adobe comes up with new updates for Adobe Flash all the time. That's because they identified vulnerabilities that third parties could use to put software on your computer that you don't want there. And that's what they're talking about, is these malware author's writing code to attack those specific vulnerabilities. And now your computer is infected with malware.

**Tags:** backdoor, botnet, certification, comptia, logic  
bomb, malware, rootkit, security, spyware, Trojan horse, virus, worm  
**Category:** CompTIA Security+ SY0-401

### **Viruses and Worms – CompTIA Security+ SY0-401: 3.1**

Viruses are well known for corrupting our operating systems and documents. In this video, you'll learn how viruses work and how worms are able to replicate without human intervention.

A **virus** is a very specific kind of malware. It's a type of malware that's designed to replicate itself, very similar to what a real virus might do in the human body. It may not even need you to click anything, but what it does need you to do is to run a program. An executable on your computer has to run, and that gets the virus going.

And once that virus get started with that executable, it can then transfer itself to other things in your computer. It can transfer to your **USB drive**. It can transfer across the network to wherever that executable is that you're executing, that you're running on a separate hard drive somewhere.

So it can go through all of your file systems. It can go through the network. And that program can really hop around to a lot of different things all from that one computer. It

can be a very, very big problem if you have a lot of different file systems you access, if you have a big network at work. These viruses tend to go a lot of different places.

Some viruses, though, you may not even know where they are. They may not do anything that's very malicious— at least, not obviously very malicious. They may be sitting there and simply making your computer run slower, or they may not be doing much of anything at all.

This is one of the challenges with viruses, is some can be very, very bad. Some can be very, very good, or at least something where you wouldn't even know that they are there. Other viruses start deleting files. They start corrupting files. They start encrypting files without your knowledge. That may be a very, very big problem.

And obviously viruses are extremely, extremely common, especially in a Windows environment. There are thousands and thousands and thousands of new viruses identified every week. That's why we mentioned that it's so important to make sure that your antivirus signatures on your computer are constantly updated. You should update then at a minimum every day so the you can be absolutely sure that if you download something from the internet that you at least have your antivirus signatures updated to identify that if it happens to be something that's known.

There are many different kinds of computer viruses. One that's been around for a long time is a boot sector virus. Don't even need operating system for this. You can sit in the boot sector of your hard drive, and when the operating system starts up, it then becomes infected.

Boot sector viruses can also be a little bit challenging to remove, because when you're in your operating system, you may not have direct access to the boot sector. So very often year after boot your machine up with a special disk or use a special program that can get access to the boot sector to be able to remove that particular kind of virus.

Program viruses are a lot more common. They're part of an application. They're embedded into an application. Maybe the virus has attached itself to the application, to the program itself, and that program virus runs whenever that application starts up in your computer.

Script viruses are things that you don't see very often, but they can still cause problems, because they are part of your operating system. Sometimes there could be scripting in a browser as well. JavaScript is a very common scripting language that is almost always enabled in a browser. And if a bad guy identifies a vulnerability in how JavaScript can communicate to other things in your browser or to what's on your computer, they can then start gaining access to your computer and doing anything they'd like to your operating system.

Another very common type of virus, and one that was really enabled by the functionality that we enabled in some of our applications, is a macro virus. When Microsoft Office first started allowing macros— so Microsoft Word, Microsoft Excel, all of the Windows and Microsoft Office applications starting enabling macros. The bad guys began to find ways that those macros could take advantage of things outside of Office. So you could run a Office program, you could open a Word document, you could open a spreadsheet, and that macro that's inside of that would then gain access to the operating system.

Now Microsoft always goes back and corrects these things. You'll see updates all the time that correct some known vulnerabilities in Microsoft Office and the way these macros work. But it's something to the back guys like to use, because so many people use these Microsoft Office applications. And the more people that are using a particular app, the more opportunities the bad guys have to take advantage of that.

Multipartite viruses are viruses that are able to use multiple methods that we've already discussed working together to do something bad on your system or to embed itself or copy itself to somewhere else. So that means that you need both a program virus and a macro virus, for example, running at the same time, working in conjunction with each other to be able to then embed or copy itself somewhere else.

Obviously, those have to be very well thought out. They have to all work together. You can't just have the program virus or just have the macro virus. They both have to be there working together to have that virus take effect.

Worms are a special kind of virus. Up to this point, we've talked about viruses being executed when you clicked on something or when you ran a program. Worms don't need you, though. Worms can propagate themselves all over the network all by themselves. All your computer has to do is be turned on. And the worm can take advantage of this.

Generally, it's taking advantage of a vulnerability that's been identified in the operating system that it then gets access into your computer, embeds itself, and then hops to another computer. Generally, when you have these operating system updates or these application updates, the things that they close are these opportunities for these worms to propagate. If you get rid of the vulnerability, these worms can't get on your computer.

But because they're using our networks to be able to move back and forth, and generally we're connecting so many systems together, they can propagate very, very, very quickly. Some worms can get on one computer in an organization, and in less than an hour, they may infest every computer in the organization.

Some worms are so good at propagating themselves that in the past, the worms themselves have created so much network traffic that they brought down the network just from a performance perspective. These days, the smart worm writers don't do that. They make sure that they can very quietly sneak around the network and get embedded to as many computers as they can so that later on, they can install a botnet. They can copy files. They can embed a key logger or do whatever they'd like to do once they're there.

A worm doesn't have to be bad, but almost all of them are. A good example of a worm that is doing something good is one called Nachi, which went out and tried to patch your computer. The problem, of course, is that even a third party program coming in the computer and making changes to your computer may not necessarily be what you want. So even if the writer of the virus had good ideas in mind and wanted to make sure that you were running the latest patch to be able to remove a particular vulnerability from your computer, even so, that may not necessarily be something that you would like to occur on your system.

One way many organizations stop those worms from coming into their environment is using a firewall or using an intrusion detection or intrusion prevention system. Those will stop the virus, the malware, the spyware, those worms, as they're coming in. And so we can stop them right there at the gateway.

But generally, that's the only place the firewall is. It's right there to your connection to the internet. That's the only place your IPS might be is that connection to the internet. You don't have multiple IPSes generally inside an organization, and if you do, they're in very limited areas.

So if one machine in your environment gets inside, gets infected, or somebody brings a laptop in from the outside and plugs it in, they can start infecting everybody in your environment with that worm. And many organizations find themselves chasing this worm down. They'll get a list of machines that are infected. They'll go out and clean those machines.

But in the meantime, those machines have been infecting others. So the next day, they're going to a completely different group of machines and cleaning those and end up spending a lot of time going back and forth trying to hunt down to resolve this problem with the worm, using up a lot of resources in their environment.

Another reason it's so hard to find these worms and to resolve getting these worms off of every computer in your organization is because the worm writers, the guys that program the malware, spent a lot of time making sure this worm would be able to propagate itself in many different ways. A very good example of this is a very recent worm and one that is relatively active— I still see this out here in people's environments— called **Conficker**. And the reason we still see it different places is because out on the internet is **a Conficker** control system that's able to communicate and provide different aspects of Conficker or back to systems.

If a computer has a shared— it's a shared computer with a weak password. Maybe there's files that are shared on that system, and it's very easy to figure out the password. Conficker will embed itself onto that computer. If you plug a USB memory stick into a computer that's infected with Conficker, Conficker recognizes this and hops over to your USB key. What it's hoping is that you'll take that USB key to another computer that has Auto Run enabled, and as soon as you plug-in that USB drive to another computer, that new machine is now infected with Conficker.

If you have a computer that does not have the latest security updates, there are many, many different variants of Conficker that tend to take advantage of these known vulnerabilities as we identify them. And as soon as a set of known vulnerabilities comes out and we patch it, another set of vulnerabilities is identified. Conficker keeps changing itself to be able to take advantage of those. And if you have open network shares, that's a great place for Conficker to just to save itself out on that open network share. The next person can go grab some files, run those files, and now they're also infected with Conficker.

So you see these guys that are developing this worm software and doing this writing have spent a lot of time understanding what's going on. And it's going to take diligence. It's going to take some technology. And it's going to take you going through and understanding how to block these things to make sure that it doesn't become a problem in your environment.

**Tags:** certification, comptia, malware, security, virus, worm

**Category:** CompTIA Security+ SY0-401

### **Adware and Spyware – CompTIA Security+ SY0-401: 3.1**

Adware and spyware is notorious for causing performance problems and application incompatibilities. In this video, you'll learn about adware, spyware, and some techniques for identifying an adware or spyware infection.

When your computer gets infected with adware, you almost recognize it immediately. Suddenly you've got tons of popups on your screen. You may be sitting there just reading a web page and three popups suddenly appear. Your eyeballs are now seeing tons of ads being thrown at it, and that's because there's usually something that's hooked into your browser or another piece of malware running on your computer, that's simply popping up ads and feeding those ads to you. Your computer's turned now into one big advertisement for many, many different things.

This can also, of course, cause performance issues for you. Having this information come across the network, this malware's probably communicating back to the mother-ship the things that you may have clicked on, the things you may have seen. There could be performance issues associated with your computer and how it's performing, once this

malware is on your computer presenting these ads, so you may see things slow down just a little bit.

This may be something that was installed accidentally. It could be something that you clicked on, and not realizing it, that that was malware. It may be presented to you as somewhat of a Trojan horse, or it may be something that's installed along with other pieces of software. It may be that the software manufacturer had no idea that these bad guys had stuck some adware along with it, and presented that to you. It was in the installation package. Or they may have included it, not realizing that it was adware.

In any case now, you've got adware on your computer, and now you have tons of popups and ads that you're viewing whatever you do. You need to be careful though when you're trying to remove the adware. There are many third party utilities that claim to remove adware that are nothing more than additional adware installation programs. So make sure that, if you're learning about that there's adware on your computer or you feel that there is adware on your computer, make sure you're using a known antivirus, anti-spyware, anti-malware program to remove that.

Either the **McAfee**, the **Symantec**, the **Trend**, the **ESETs**. The well-known antivirus manufacturers will be able to give you software that you can trust to be able to remove those things, rather than trusting some third party popup that may have appeared that says you have adware, and now you would like to remove it, [click here](#). That's definitely not the way to go about doing things.

As the name implies, **spyware** is software that is specifically designed to watch what you're doing. It's spying on your browsing. It's spying on what you're typing in at the keyboard. It's trying to identify a lot of different things about you, and that's because the software these days can present advertising that's tailored to you. It can provide private information back to someone else that can then use that for identity fraud.

These days there's really, really big money in getting your identity, and getting your private information, so that people can open up other lines of credit, credit cards, open up bank accounts with your personal information, or even worse, go into your existing credit cards, and your existing bank accounts, to gather the money directly from you.

Usually these trick you into installing. It thinks that there's fake security software. You may see this advertised all the time, about installing this anti-spyware onto your computer, that really is spyware. Peer to peer networks tend to be a very, very large area where people will embed spyware, present an executable file. Or say that there's a brand new piece of music that's available that you download, and unfortunately, now you have spyware on your computer.

Browser monitoring is a big part of spyware. Identifying your surfing habits, finding out where you're visiting, especially if you happen to be visiting a bank site, a health care site, somewhere where your personal information might be available. There's juicy tidbits and details they can find out about you. They're going to track where you go and find out how you log into those.

One problem with these pieces of spyware is very often they include key loggers, so they're tracking everything you type in. You can then have a big file of all the things you typed in for the day, and that file simply set off to the mother-ship. Behind the scenes, you never even though the file was transferred out, and that's now somewhere else on the internet that contains login names, passwords, and anything else you may have typed in that day.

There is a lot of adware, and a lot of spyware out there, and there's three big reasons why you see so much of it out there. The first reason is money. Your eyeballs are very valuable. Your buying habits are very valuable. If we can get advertisements in front of you, and I can present someone else's ads, I'll have a third party paying me to present

those ads. So if I infect a million machines, and I'm presenting a million advertisements to people, that's big money for me. And if I can do that by embedding adware onto your computer, it may not be something that is legal, but it is something where third parties are making money off of your eyeballs.

Another reason is money. Your computer, the bandwidth to your computer, the time on your computer is very valuable, and if we can embed spyware or adware or other pieces that will run onto your system, that becomes very, very lucrative for somebody. And your computer time and your bandwidth is something that, if you distribute across a million computers, it's much easier to compile a bunch of strangers computers to do the things we want rather than using a million machines that I might buy and stick in a data center somewhere, very, very valuable.

And the third reason is money. The information that is on your computer is important, but your money is very, very important. And if I'm tracking your logins to your bank account, I'm tracking logins to your credit card information, I can access those accounts directly and start transferring money in and out. I've now got a lot of control over your personal dollars.

Even worse, is when this spyware and this malware gets onto your computer, tracks your keystrokes, if you're at a business. And as a business, you're in the finance department and you have direct access to your bank account, you might have thousands of dollars, tens of thousands, hundreds of thousands, millions of dollars available in your bank account, and now sending that username and password information back out to some third party gives them access to large sums of money. And this is a very, very big problem with spyware being embedded on our computers today.

It's very important, therefore, to keep your operating system up to date, to keep your applications up to date, keep your anti-spyware and anti-malware signatures up to date. And generally, if you can keep all of those things up to date and follow very good procedures for not clicking on things inside of your emails and not clicking on unknown links in a web browser, you can generally stay very, very safe and avoid having the adware and spyware installed onto your computer.

**Tags:** adware, certification, comptia, security, spyware

**Category:** CompTIA Security+ SY0-401

### **Trojans and Backdoors – CompTIA Security+ SY0-401: 3.1**

Trojan Horses are a special kind of malware that manages to infect our machines by tricking us into running the malicious software. In this video, you'll learn about trojans and backdoors, and I'll demonstrate how an application posing as a game can quickly infect our computer.

A **Trojan horse** is a unique kind of malware that is able to sneak onto your computer to do the things that it wants to do. The name comes from the historical use of a Trojan horse. This is what the Greeks built. They built an enormous wooden horse and put it outside the gates of Troy, and when the Trojans found it, they pulled the horse inside. And once it was inside and night fell, the Greeks came out of the horse and they were inside. They didn't have to now get through the gates of the city, and they were able to conquer Troy just by sneaking in in this enormous wooden horse.

This is the idea that we have in Trojan horses on our computers, except it's a digital type of Trojan horse. It's one that sneaks on to your computer pretending that it's something else, so that they can then get inside. Once it's inside your computer, you've given it rights to run the program that it's running. You said absolutely you can run. It's a video from somebody? Yes, I'd like to see that. Oh, this is a program that shows me a nice winter

card that I can see or a spring card, or somebody sent me a birthday wish. I'll open up this program to see what it is.

In fact, it may even show me a birthday card. It might even play music for me. It may show me a snowy wooded evening. Unfortunately behind the scenes, it's also embedded itself.

Its primary purpose is to get on to your computer. And it's not entirely concerned with replicating. This isn't a virus or a worm. Its job is to get on your computer and to fool you into allowing it into the gates. It's getting around your existing security.

And as soon as it gets on your computer, you'll find one of the first things that the Trojan horses do these days is disable your antivirus. So if you happen to notice that your firewall isn't working anymore on your computer and your built-in antivirus isn't working anymore, it's very, very possible that you've been infected with a Trojan horse, because now it can do whatever it likes. Because your firewall is turned off, it can now open up some back doors and let other devices in. Now that your antivirus is turned off, it can download other things on to the computer that may be malicious and be able to do other things, embed keyloggers, and do other types of methods of infecting your computer, because there's no way to stop it now. It's disabled your antivirus. It's disabled your firewall.

And it's those back doors that create such a problem for us. Once we get that one piece of malware on our computer. These malware manufactures, the writers have realized that once we get on a PC, this is great. We can open up a back door. We'll go around the back of the house and just unlock the door and allow whatever we would like in through the back door of your computer.

And that way, they don't have to worry about finding another vulnerability. They don't have to worry about finding a way to authenticate properly to your computer. They've already got that first step in, and now that they're inside of your computer, they'll simply open up a back door around back and then put whatever they would like on your computer, add additional malware, additional spyware. And that's why whenever you find a machine that's been infected, it isn't just one thing. It's multiple things, because the malware manufactures, once they find that opening, they absolutely take advantage of that.

There is some software that includes a back door with it. And this isn't something that is very common, but something you need to be aware of is that there was actually a Linux kernel version that's somebody wrote a back door into. And it was one that was found very quickly. Thanks to open source, people were able to go through the code and say that looks funny. Why did somebody happen to put that in there? That gives them access into anybody's Linux machine.

There's also bad software. As part of the application, a manufacturer may have maybe not intentionally created a back door, but nonetheless found a way to have something there that they can access machines that had that software on them. And unfortunately, that is also a big problem. That's why whenever you install new software onto your computer, especially in a large environment, there's tests that you can do with that computer to make sure that it is as protected and secure after installing that software as it was before installing that software.

I recently had one of my computers infected with a Trojan horse. And so I collected that Trojan horse. I put it into a digital Mason jar and put it in the corner. I've got it running in this virtual machine. So this is not something you'll want to do on your computer. I've taken and put it into a test environment. And now what I'm going to do is run it and show it to you. And when we're done with this, I have a snapshot taken of this virtual machine. I'm going to revert back to a previous date and time, which essentially erases everything that we're about to do here so that I maintain this protected system.

This Trojan horse that was found, I'm going to look at the properties of this. It's called **GBT**. And you notice, **Games for Windows Live splash screen**. It's a game. Who wouldn't love to put a game on their computer? I'll absolutely run that program. That sounds like a great thing.

And what you're going to find is when that program executes on this computer, what you'll notice is, first thing that pops up is your Windows security center. Your firewall is turned off. Your anti-spyware and anti-malware is turned off. And then another window pops up called **XP anti-spyware**. Well, what could be bad with anti-spyware? But oh, no, it's identified programs on my computer that are infected with spyware. Here's a worm. Here's other spyware on my computer. It's finding all kinds of infections on my computer.

Now the reality is, what we're running on right now is a stock installation of Windows XP. There's no additional programs that have really been installed here. What this is telling us is absolutely fake. None of these things are real. And because it used the Trojan, pretending it was these games, now it has disabled my real firewall, disabled my real antivirus. It's created a back door and now it's presenting to me this front end that's telling me that I have all kinds of viruses, macroviruses, viruses, botnets, and a lot of different things on my computer, none of which is absolutely real.

This is a new type of malware that we're seeing called scamware or ransomware, because at the end of this what it's going to say is that I can absolutely remove this from you. You need to give me some money and I'll be able to remove all of these problems from your computer. But of course, none of these problems existed to begin with. This is simply scaring me into providing a third party with credit card information or providing them with money in some way, shape, or form for doing nothing but embedding a malware onto my PC.

There, it's now finished its scan. It says attention, danger. It found 26 critical system objects that were infected. Well gosh, I should probably register this so I can get rid of that. Obviously don't do this. You don't want to register this. And it pops up this very, very **professional-looking XP anti-spyware** I can buy now. There's frequently asked questions. One-year licenses only \$60 United States funds to be able to remove that. And look, they even have a Like button for Facebook. One million people like them. Well gosh, I'll like them too. Let me click that. Doesn't actually go to Facebook.

That's a fake Like button that's there. But doesn't it look real? Follow us. Join the conversation on Facebook. It's not going to let you to Facebook, to their page, because their page doesn't exist. This is an absolute scam. None of this is real. All they want are \$60, \$70, or \$80, to be able to get that money. And at the end of that, it may or it may not enable or disable this particular malware on your computer, because they don't care. You can't contact the company. You can't get your money back. At that point, you are infected all because that Trojan presented to you that it was something like games, and that sounded great to me. And now my computer is really infected with this malware.

It becomes now, very, very difficult to remove this from a computer. It has embedded itself into the operating system so that any time you run a program, it will run and make sure this Trojan is running, which means you can't simply remove the file that you happen to find associated with the Trojan. It's hiding it in some temporary directories. It has now embedded itself in multiple places. If you turn off your computer and turn it back on, it's going to start up automatically. And it's going to constantly pester you that your system has been hijacked and security threats have been detected and you need to give them some money to remove these things.

Very often, what we end up doing is simply removing and taking off and backing up our personal files, our documents and other things where malware can't be infected, securely backing those up somewhere and then completely nuking this hard drive, erasing everything on it and reinstalling the operating system from scratch. Very often that is the

only way that you can be 100% sure that you have absolutely removed that Trojan from your computer.

**Tags:** [backdoor](#), [certification](#), [comptia](#), [malware](#), [security](#), [Trojan horse](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Rootkits – CompTIA Security+ SY0-401: 3.1**

One of the most significant challenges with rootkits is their ability to be invisible inside of your computer. In this video, you'll learn where rootkits live and how they manage to avoid our normal malware detection strategies.

Rootkits come from a name that is based on something you'll find in **Unix-type systems**. If you are the super user of a Unix system, then you are root, very similar to being the administrator on a Windows system. And that's where the word comes from. If you have a kit of software that allows you access to that machine, that gives you root access, then that is a rootkit.

It is something that is trying to hide itself. And a good way to hide itself in a computer is to become part of the operating system itself. And if you can become part of the kernel or embed yourself deeply in the inner workings of an operating system, then you've got a lot of power. And you can circumvent normal security on a system. That's not very easy to do, of course, but if you are able to accomplish that, then you've got a lot of power on that particular computer.

This means that it is invisible to the operating system. A **rootkit** is something you're not going to see through normal means. You're not going to be able to pop up your Task Manager and be able to see it there. You're also not going to be able to identify it through normal anti-virus utilities, anti-spyware utilities, anti-malware utilities. They can't see it. And if they can't see the malware, they can't see the rootkit that's on your system, then obviously they can't remove it, either.

Sometimes you can hide yourself in an operating system, just by blending in with everything else that's there, doesn't have to be a very complex process. For instance, in Windows operating system, in the Windows System Directory, there's very commonly thousands of files inside of there, hundreds of megabytes of information. Just drop a file in there. Who's going to even realize, in that thousands of files, that you've now added a new one to the mix. Especially if you're very sneaky and you name it something that looks legitimate. If you name it, run32dl1.dll, in this particular font it looks pretty obvious. But if you look at it in your Windows frontend, that 1 looks just like an l, which means it looks just like run32.dll, which is a very, very common dll on a Windows computer.

So by sneaking the name in there that looks very, very familiar, you can glance at it and miss it, especially if it's in a directory of thousands of files.

One of the most historically notable rootkits was one that was created by **Sony BMG**. This is the part of Sony that creates music. And they distributed a music CD that had ' capability to put it in your computer. And obviously your computer is able to play CDs. You can see this is in 2005, when CDs were still relatively popular. You put the CD in your computer and the music would play. But, behind the scenes, Sony installed a rootkit on your computer, obviously without your permission. This is something not everybody would want. But Sony was trying to protect people from copying the music. And they obviously were using a very bad method to be able to control those things.

Anything that had a dollar sign sys, dollar sign period in front of it, was completely hidden from the operating system. And that's how Sony hid their software on your computer. Unfortunately, of course, just by naming a file and making it hidden in this way, meant that other bad guys could do exactly the same thing. And they did. Once this was identified

as having this problem, the bad guys went into overdrive and said, you mean we can hide from the computer? We can hide from your anti-virus? We can hide from the anti-malware, just by naming a file, dollar sign sys, dollar sign period, and then anything? Yes, absolutely you could. And so that happened very, very quickly.

Well, once Sony was presented with this and with a lot of people complaining about it, they issued a patch. Unfortunately, the patching process opened up a back door in the computer that potentially allowed other malicious software to get installed. So, really, just a bad situation all the way around. It was badly created. It was badly solved. There was a lot of problems with this. Ultimately, this was only a couple of months later, Sony said, you know what? We are recalling every one of these CDs. And if you bought one of these CDs or you were infected with this, we're going to give you money, and give you the opportunity to download some music for free, so that this entire problem could be resolved to everybody's satisfaction.

The **Sony BMG rootkit** was one that was identified by Mark Russinovich, who is the guy who created, one of the guys, who created Sysinternals. We had Sysinternals.com that was acquired by Microsoft. So he works for Microsoft now. But he has a product called RootkitRevealer, absolutely free to download from Sysinternals, that can tell you a lot about the things that are in your computer and things that may be hidden from your operating system. Uses some additional capabilities to find those things. So you can go to the Sysinternals area of the Microsoft website, download RootkitRevealer. And there are a lot of forums and other conversations you can participate online with, to discuss the things that you're finding when you run RootkitRevealer on your system.

There's also ways to remove rootkits, using very specific removal software. Because root kits may use a very specific method to get on your computer, you very often have to have a very specific uninstaller for that rootkit. So once somebody discovers the rootkit, they see where it's embedded, they understand what it's doing, it is generally a lot easier, at that point, to create something that can remove that rootkit, once and for all.

**Tags:** [certification](#), [comptia](#), [rootkits](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Logic Bombs – CompTIA Security+ SY0-401: 3.1**

Logic bombs can be very destructive and can reside in our systems with near invisibility until they trigger. In this video, you'll learn about common logic bomb categorizations and some real-world examples of logic bombs.

Logic bombs are types of malware that are waiting for something to happen. They're waiting for this pre-defined event to occur. At that point, something goes into effect. Files are removed, systems are rebooted, other things are deleted, systems are corrupted. There could be many things that happen with a logic bomb.

Very often these logic bombs are left by people who have a grudge. It's someone who's been dismissed. They've now been fired from the company, but before they leave, they're going to set this bomb ticking. They're going to set a program in place that once they walk out the door and days or weeks or months later, something may cause problems inside of that organization. So obviously it's something that becomes a very, very big issue. This can be a date or time that occurs, and when that happens, the logic bomb goes off.

Or maybe it's something that happens with users. Maybe is a file that is added to a computer or removed. Maybe it's the next time a system reboots is when this logic bomb goes into effect. You're never quite certain until you find the actual bomb to understand exactly what might cause it to go off.

These can be really, really difficult to find. Obviously, they're not a virus. It's not something that's known by anti-malware or anti-spyware, and if it goes off, the people that are writing these logic bombs are generally destroying things. They're destroying files, they're creating corruptions inside of operating systems, they're making a life really painful for everybody else. And so once the bomb goes off, it can be very, very difficult to recover from that.

Unfortunately, there have been some very well-documented cases of logic bombs. And if you go out to Google and you search for some, you can see all kinds of news articles. Here's a couple of good examples of one. This one was at Fannie Mae, so a very, very large organization. He set, this is someone who had been dismissed by his job that set a logic bomb to completely disrupt over 4,000 servers at their organization. Now in this particular case, fortunately the logic bomb was found before it went off and so the entire script that was built to really create problems never really created a problem for the organization. Obviously, though, there was still legalities involved. There was still a prosecution, and there was still penalties associated with that.

Another example of a logic bomb, this is another large organization, UBS, where the system administrator was fired and then put a logic bomb onto the systems. And one of the things that he did that made this one especially bad is after he put the bomb that was going to take out a huge part of this organization— this is a bank, a financial organization— he went to a stockbroker and got put options, which means if the stock went down, he would make money. So obviously this is a very, very big problem.

They found this one before it went off as well. So in this particular case, they avoided a lot of problems, not only in their organization, but a lot of problems with what could have been stock fraud and things that should not have occurred in our financial system. And again, there were legalities involved, and we run into this a lot, not necessarily with spyware and malware, but certainly with hacking and other types of malicious things that might occur is that there are legalities involved. So you want to be sure that if you're dealing with any type of situation where somebody has left a logic bomb in your environment that you're handling it in the ways that we discussed in some of our previous videos.

**Tags:** [certification](#), [comptia](#), [logic bomb](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Botnets – CompTIA Security+ SY0-401: 3.1**

The remote-control of a botnet has the ability to cause significant harm to our computers and our assets. In this video, you'll learn how botnets work and how one of the largest botnets in history has become very good at taking money out of our bank accounts.

A **botnet** is a type of malware. The name comes from robot networks, and this is one where your computer has now been infected and you may not even realize it. But behind the scenes, there is a robot living there in your system that is under the control of someone else. A third party can now have your system send spam. A third party can have your system participate in a denial of service. Maybe it's simply sitting there and sending your private information out to that third party. A lot of different things can happen, and you may have no idea that botnet is on your computer.

It probably got there because there was a Trojan horse, you clicked a link somewhere, maybe it was something in an email. Maybe you thought you were installing an absolutely legitimate program, but unfortunately it was installing a botnet on your computer. Sometimes you don't even realize it. It may be a worm that takes advantage of a known vulnerability or even an unknown vulnerability with an operating system or with an application, and now the botnet is living on your PC.

The bot's very simple process is it sits there and waits for commands to come in. It's connected to the internet. It checks in with the mothership, sees if there's anything waiting. It may open up a back door or two and simply go into a listen mode and wait for the directions to come from the mothership. And then that central computer sends information down to your computer that says OK, grab the key strokes, or participate in the botnet, or send this spam and then your computer goes into effect and does all of the things that the mothership tells it to do.

One of the most fraudulent botnets in history, and I say financially fraudulent botnet in history, is **Zeus**. This is a piece of information I gathered from the **FBI**. If you go to this website **FBI.gov**, you'll be able to find this diagram along with a lot of other information about how **Zeus** really works. And what really happens behind the scenes is you have somebody going through and creating the malware itself, creating the botnet. And they hand that off to the hacker that finds a way to get it to your computer, either through a malicious link or they find a vulnerability or they embed it in a worm, but they get it onto your computer.

And once they're on your PC, the whole purpose of **Zeus** is to get your banking information. That's the whole reason it was written. It doesn't care about all the other things you do. It doesn't care about the Facebook and the Twitter and all the other logins you have. It wants your money. So it then gets your banking information, logs onto your system, and then starts going right out to your bank and taking your money away from you.

It usually sends that money to a third party, to a mule, whose job it is to get it from one place to another. Because once they get your money, they need to hide it in some ways so it may go to a bunch of mules who then get the money back to the organizers and receive millions and millions and millions of dollars just by embedding this botnet on your computer. Unfortunately, **Zeus** continues to be popular.

There are many different variants of Zeus that use many different ways to get onto your computer, gather the information, and provide it back to the mothership. This is a graphic that comes from [zeustracker.abuse.ch](http://zeustracker.abuse.ch), and it shows you how many active Zeus files are out there, how many online binaries, how menu drop zones there are. And you can see as these servers, these central machines go up and down, you can start to see exactly how much is going on out there on the internet. So if you're wondering if Zeus is something that might be affecting you, it's affecting a lot of different people today. This is something to keep your eye on, not just with Zeus, but other types of botnets as well.

**Tags:** [botnet](#), [certification](#), [comptia](#), [security](#), [zeus](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Ransomware – CompTIA Security+ SY0-401: 3.1**

Ransomware is a specific kind of malware that goes directly after your wallet. In this video, you'll learn some of techniques that ransomware uses to get your money.

A type of malware that's getting very good at removing people from their money is **ransomware**. This is where the bad guys want your money, and they're going to lock down your computer until you give them exactly what they want. In some cases, this may be actually a fake ransom like the one we see here where you've got a Federal Bureau of Investigation label.

It tells you that you've been going to inappropriate websites, that you have inappropriate information on your hard drive. It even tells you that spam messages with terrorist motives were sent from your computer. And of course, none of these are probably true. But they're scaring you now into saying that you have a big problem that you can simply solve by sending them some money, a fine of \$200. That actually doesn't sound very much for all of these bad things that I did, but who am I to argue? I'll simply go to one of these locations, put together a money pack, and send the bad guys the money pack.

This is becoming a very effective form of malware because it's scaring people enough who really don't understand what they're seeing, and they're sending the bad guys the money in the hopes that that will then remove this particular warning from their computer. This type of warning and these messages may be something that could be easily removed, however. You'll need to take a system to a trained professional or be able to find out more about this specific form of ransomware to know whether it's something that can be easily removed or not.

The newest generation of ransomware, however, cannot be removed easily from your computer. In this particular case, the bad guys are encrypting the data on your computer but leaving your computer able to work properly except now all of your data is no longer available to you. They are encrypting every single bit of your personal files and leaving the entire operating system intact. This is because they want the operating system to continue to operate normally. They want you able to go through the entire payment process so that you can then send them the money that they're wanting.

And at that point, ideally, the bad guys are going to send you the decryption key. This is often combined with a countdown timer. So unless you're able to send them the money within a particular time, all of your data will be inaccessible to you forever. This is using public key cryptography. So this is not something that can be easily decrypted. In some cases, if you don't want to pay this particular fine, you would pretty much have to delete everything on your computer and restore from backups. If you don't have backups, then you probably have a difficult decision to make. You either send the bad guys the money that they're asking for or you decide that you're never going to see this data again.

**Tags:** [certification](#), [comptia](#), [ransomware](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Polymorphic Malware – CompTIA Security+ SY0-401: 3.1**

Polymorphic malware is constantly changing, and this makes it difficult to identify and remove. In this video, you'll learn about polymorphic malware and the methods that we're using to control and eradicate this malicious software.

Before we talk about polymorphic malware, let's discuss the ways that we identify malware today. In many cases, the technologies that we're using to find malware on our networks and on our computers is done using signatures. These are very static pieces of

information. And we're simply looking to see if we can match this information with what happens to be going across our network, or what happens to be executing in our systems.

If we identify a match with these signatures, we say that that is probably malware. And we remove it from executing on our computer. Or we remove it from the network. Many malware detection engines use signatures, but also use technologies called heuristics.

Heuristics are looking for a certain event to occur. They may be looking for a system file to be changed. And if that system file is changed, the heuristics may determine that this executable is malware. In these cases, you don't even need a signature. You're simply looking for a particular kind of event to happen.

As you may have guessed, heuristic-based detection requires a number of additional resources. You have to have something in memory or executing to be able to identify what's happening on a system. And it has to be looking at many things all at the same time. In very large scale implementations and very high speed networks, this becomes almost impossible to do.

**Polymorphic malware** is designed to take advantage of the problems associated with signature-based malware detection. Polymorphic malware will change itself every time it is downloaded. So when one person goes to a website and downloads an executable, and then the second person goes to the same website and downloads the executable from the same link, they actually receive two different files.

Obviously, inside of the file is the attack code. And that didn't change. But everything else around that attack code did change. This creates problems for those signature-based detection engines, because the signatures are looking for one particular kind of data.

Another method the malware authors use is to encrypt the attack code. And they use different keys every time. This means that the attack code on one system will have a completely different signature than the attack code on another system, even though it's exactly the same attack code. Only after decrypting the attack code do we see that it is exactly the same on both systems.

With polymorphic malware, there is still going to be part of the executable that is exactly the same. And the signature detection engines are going to take advantage of this by trying to find exactly the piece of that malware that is the same, regardless of what changes around it. In this way, we're able to create a single signature, but hopefully affect and identify many, many variants of exactly the same malware.

And, ultimately, the only way to identify some of this polymorphic malware is through the use of heuristics. But, again, that's difficult to apply on a very wide scale. And it's going to use more resources inside of our systems.

**Tags:** certification, comptia, malware, polymorphic, security

**Category:** CompTIA Security+ SY0-401

### **Armored Virus – CompTIA Security+ SY0-401: 3.1**

The virus developers know that the secret to remaining active is to hide as much as possible. In this video, you'll learn how virus programmers use obfuscation to create armored viruses.

The malware authors and the anti-malware authors are in a race with each other. The malware authors want to get their software distributive on as many systems as possible without being infected, and the anti-malware authors want to be able to create protections so that they're able to stop this malware as quickly as possible. One thing that the malware authors do is try to obfuscate or make their code a little bit harder to understand by creating an armour around they're malware.

One of the first things anti-virus and anti-malware authors do is they tell their code inside of their executable to jump other places should something start scanning. In this way, they're able to take the anti-virus scanners and have them go elsewhere rather than looking at the actual code of their virus. Ultimately, the anti-virus researcher is going to identify this executable as something that is malicious and they're going to deconstruct this code. They effectively disassemble it so they can view the actual machine code used by this virus and they start examining exactly how it works. The virus author knows this is going to happen. So they've added obfuscation or they've made the code more confusing by adding unnecessary code or nonsense code. And the researcher still has to go through all of this nonsense code to try to determine where the real virus is and where all the obfuscate code might be.

This is where the race really happens because as long as a signature is not created for this virus or this malware it can continue to be installed on everybody's computer. The anti-virus researcher of course wants to create a signature and get that out to everybody who's running their anti-virus software. The longer it takes for that anti-virus researcher to find the code and create a signature, the longer that virus is going to be active out in the wild.

**Tags:** [armored](#), [certification](#), [comptia](#), [security](#), [virus](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Man-in-the-Middle Attacks – CompTIA Security+ SY0-401: 3.2**

One challenge with modern network security is protecting against man-in-the-middle attacks. In this video, you'll learn about man-in-the-middle and I'll demonstrate a live **MiTM attack**.

A man in the middle is an attack technique that works very much like the name sounds. There's a bad guy. He sits in the middle of a conversation between two devices. And he's able to watch exactly what's going on between those systems.

He can capture packets. He can inject his own information in there. He can change information or simply just watch what's going on and see if we can identify things that might be interesting that he could use later.

What he's really doing is redirecting your traffic. He becomes the endpoint instead of you sending information, for instance, to your router, you would send information to the man in the middle, and the man in the middle would then pass it on to the router, in many cases being completely invisible. You never even know this redirection is taking place. So obviously, this becomes a pretty major issue.

There's a number of ways for accomplishing this. One very common way is something called **ARP poisoning** because the ARP protocol, the address resolution protocol used in **TCP/IP**, has no security associated with it. Machines just simply trust that if they're receiving information in an **ARP** packet that that information is something that is legitimate.

And what we're going to do is use a program, and I'll show you exactly how we're able to use third party applications to be able to do ARP poisoning with just a few clicks of a mouse. What we're going to do is sit between two computers. For instance, you'll have a 192.168.1.9, which is one of the Macintosh computers on my network. And I'm going to be talking out to the internet through my router, which is 192.168.1.1.

You'll also notice that my laptop or my computer I'm using here has a particular Mac address on the ethernet network as does my router. And then fundamentally, of course, we don't use IP addresses to communicate directly between devices. We use Mac addresses to communicate directly between devices over ethernet. And then once we send the traffic, the IP address helps move it along wherever it wants to go.

The way that we determine where this router is though is we send out a message wanting to know where is 192.168.1.1? Are you out there? I need your Mac address so that I can send information directly to you. And what happens is the router in this case says, oh, yes, I'm 192.168.1.1 and here is my Mac address.

And what my machine over here does is make a little ARP cache that is inside the memory of my computer, and it says, oh, I can remember this now. I don't have to keep asking over and over. I'll always know that 192.168.1.1 happens to be this Mac address and all is well with the world.

Well, what happens with a man in the middle situation is you have somebody else who's on this local network. And that's an important consideration is ARP and the way that it operates and these man in the middle attacks that take advantage of this ARP poisoning can only work on this local subnet. ARP packets do not traverse routers. So this is something that, obviously, has to be on our local network, which is another reason to have your wireless networks well secured.

So there's a third party out here, 192.168.1.14. His Mac address, as you see here, I made it very easy to find his Mac address. ABCDEF. So what he does is he sends a non-solicited message over to 1.9 saying oh, I'm 192.168.1.1. My Mac address is AABBCCDDEEFF.

So what happens is that the machine out here on 1.9 says, oh, it's changed? I had no idea it was different. Let me change my cache so that now I will always send this ARP cached information. There it is. Now whenever I need to talk to 192.168.1.1, I'm going to send it to this third party.

And what happens is the third party then sends it on to the router, and now he's in the middle watching everything that goes back and forth. He poisons 1.9. He poisons 1.1. And now he can watch everything go across the network.

Let's see this ARP poisoning in action. What I have here is a virtual machine running Windows. And on this machine, I'm running an application called Cain and Abel. You can access this at [oxid.it](http://oxid.it) if you wanted to download this and try it yourself.

Now also, this is my Macintosh computer and I'm going to look at the ARP table, ARP-A. If I'm communicating to 192.168.1.1, I've already done an ARP and I have in my ARP cache this Mac address, which is the Mac address of the router that I use to communicate out to the internet. So any time I want to go there, I simply talk to that particular Mac address.

Now what Cain and Abel is going to do for us it's going to poison that ARP so that we can communicate through Cain and Abel to get to the internet. Let's see how this works.

I don't know the device is on my network now. I'm going to turn on the sniffing function of Cain and Abel. Go to the sniffer tab for the host, and I'm going to list the hosts that are on the subnet by taking the default settings and clicking OK. I've got a number of devices on my network. The one I'm most interested in is this 1.9 so you do need to know the IP address or perhaps, the DNS name of the device that you would like to access to be able to do the poisoning.

What I want to do is go to this APR tab and that stands for our ARP poisoning functionality. And what I want to be able to do is click this plus sign up here and say, anything going from 192.168.1.9 going to 192.168.1.1, I'm going to click OK and tell it that's the two that I want to poison. Those right there.

Now the Cain and Abel program can poison many devices simultaneously so that you could start up Wireshark and capture information or do some interesting things that are built into using Cain and Abel. Now it's not doing the poisoning yet. What I have to do to poison is click that nuclear symbol right here at the top it says right here, poisoning. And it shows you how many packets.

And if I do an ARP-A notice that 192.168 is now AABBCCDDEEFF, which is the Mac address of this device. So it's now sent also a poison to my router so that it knows where to get back to 1.9 is it should go through Cain and Abel. Now we want to test this.

So let's try going out to the internet and see what we see. I'm going to bring over a browser screen. And I'm going to connect to my internet router. That will be a good way to tell.

Since I'm starting on 1.9, I'm going to go to 192.168.1.1 and hit Enter. And it's going to ask me to log in to this router. I'm going to put in my password and I'm going to click Log In. And it's going to log me into this Netgear router.

Now because we had a man in the middle, this man in the middle was able to see all of those packets. And I could have certainly gathered them, but there is an automatic function within Cain and Abel that will grab all of the passwords that go through **FTP**, **HTTP**, **IMAP**, **LDAP Pop3**, **Telnet**, et cetera, et cetera. It's already identified a session that went to 192.168.1.1 with the username of admin and the password of supersecret. So just by doing this poisoning on this local network, I'm now able to watch everything going by and I'm able to gather some very, very critical security information from the traffic going back and forth over my network.

That's another good reason why we want to be sure if we're communicating to a device that is very, very secure that we want to use encrypted protocols to be able to do that. I'm going to turn off the poisoning. That gets rid of that poison there. Let's go back to our terminal screen.

I'm going to do an **ARP-A**. When I remove the poisoning, notice everything went back to normal. So Cain and Abel is able to do a man in the middle, slip in the middle, watch what was going on, and then quietly remove itself from the middle and everything goes back to normal.

**Tags:** certification, comptia, man-in-the-middle, MitM, security

**Category:** CompTIA Security+ SY0-401

## Denial of Service – CompTIA Security+ SY0-401: 3.2

Denial of service attacks are very difficult to defend against. In this video, you'll learn about denial of service attacks and you'll see how one of the first DoS attacks, the Smurf attack, was able to disrupt services on many networks.

The most basic definition of a denial of service is when you are preventing a service from operating. And this can be done a number of different ways. In the computer and networking world, there are a lot of different ways to create a denial of service situation. If you are to overload a particular web server, for instance, with thousands or millions of people hitting it all at once, it would cause that server to be completely overwhelmed by the number of requests it's getting and, therefore, not be able to provide services to the people that would really like to, legitimately, get access to that web server.

Denial of service is also designed to take advantage, sometimes, of a very specific vulnerability. For instance, there may be a router. When a router receives a particular kind of packet, it doesn't know what to do with that packet. And it causes a problem inside the software of that router and may cause the router to stop forwarding traffic. This is something that many router manufacturers have had to deal with. And whenever they find a vulnerability like that, obviously they patch it very quickly.

But the bad guys know about these vulnerabilities. And if they find a router that isn't patched, they may be able to find that particular design failure or that particular vulnerability and take advantage of it, completely bring that router down. Now that router has to be rebooted for it to be able to work properly.

You may also just want to cause an entire system to become unavailable, not for a malicious reason, necessarily, but more of a competitive advantage reason. We see this a lot in industrial espionage, when there is one particular organization that would like to keep their competition out of business. And they may do that through nefarious means, such as creating a denial of service situation.

This may also, of course, be a smokescreen for another kind of exploit. In a previous video, we talked about a man-in-the-middle attack. We talked about **DNS spoofing**. We've talked about **DNS poisoning**. And when you start to do those types of things, it may become very useful for you to create a diversion or to overload a legitimate server, so that you can take advantage of that. And you can become the legitimate server. You're now available, that other machine no longer available. You're now the new DNS server. And when it comes to DNS poisoning, that can be very, very, very useful, very helpful when you're performing that type of attack.

This, of course, does not have to be a complex process for a denial of service. I mentioned the very basic definition of a denial of service is to prevent any access to that resource. One good way to do it would be to turn off the power. You don't have to have a million systems hitting a web server at once. You simply go outside the building. You notice outside of the building the power, the big switch for the power system on the building, is not locked up. You walk up. You turn the switch. The entire building goes down. You've just created a denial of service situation.

When we're talking about big time denial of service, though, we're talking about many devices on the internet, participating all at once. Wherever we were looking at botnets, in our botnet video, we were talking about these botnets just sitting there and waiting for a command. And the bad guys may take an army of their robot network devices out there, send them a command that says, please take down a web server, send a denial of service attack to a particular site. And so you can use this army of computers to bring down that service. Use all the bandwidth, use as many resources available in that web server. If that

web server can serve 1,000 users at once, hit them with 2,000. He'll be so busy trying to serve that many systems, he won't be able to operate and do anything for anyone else.

This is exactly the purpose of a botnet, not just to send spam, but to participate in these massive types of events. Coreflood was a really good example of this. This was taken down in April 2011. 2.3 million devices, it's estimated, were participating in this Coreflood botnet. And these botnet command and control would send these devices out to take down individual systems. Becomes a bit of an issue when you're trying to keep those websites up and running. There's really very little you can do when that many systems are hitting you all at once.

This may also be something called an asymmetric threat, which means the attacker, the individual device, has fewer resources than the victim has. Sort of an odd situation. You've got all these individual boxes out there. They may be very tiny devices. But they are now all working together to take down a much larger giant. That becomes a real big issue. It's very difficult to build a web server that would be able to accommodate 2.3 million requests all at once and, therefore, it's very, very simple for these tiny, tiny little devices to take down those monster websites.

One of the very first denial of service attacks was one called a Smurf attack. And what was nice about the Smurf attack is you could get a lot of bang for your buck. And here's what I mean by this. You would be a station out on the network. When the 192.168.1.22, let's call you on this laptop, and you would like to take down the server that's at 192.168.1.1. But one machine that's sitting out here, it would be very, very difficult—because this is a big beefy server—it would be very difficult for one machine to take down that one server.

So the key to a **Smurf attack** is that you get to involve everybody else on the network. And the way you do that is through something that was very commonly done back in the day. This was back in the '90s. This is a capability that, really, you don't see much anymore. But you send a ping out, an ICMP echo request. Then you spoof the From address, even though you're 1.22. You send the packet out and say, Hi. I'm 192.168.1.1. You pretend you're the server. And you send it to the broadcast address for the subnet.

When all of these devices out here see this packet, they are all programmed to take in and react to broadcast addresses. Everybody must look at a broadcast frame. That's the whole purpose of a broadcast. And they look at that and say, wow. That's a ping. I need to now send this ping response back to the 1.1 address. And now everybody on the network sends a response back.

So by sending out this one packet, you can get an entire subnet sending back ping commands. And if you start streaming out those ping requests, obviously you'll start multiplying the amount of ping responses going back to that individual server. And, hopefully, if this is your objective, you're creating a denial of service situation for that server.

Now, the reality is that, today, our routers are not going to route these subnet requests, these broadcast requests, for a ping to the subnet. And our individual workstations these days are programmed not to respond to a request. You'll still see one from time to time that won't participate that way. And that may be an older system. Or it may be a type of operating system that will respond to that, but generally, you'll find most systems won't participate in a Smurf attack. But it was a very, very common way, in fact, one of the very first ways, to do a denial of service and make it very, very easy if you're on that subnet, to take down one of these big servers, all with a single packet.

**Tags:** certification, comptia, ddos, denial of service, DoS, security, smurf

**Category:** CompTIA Security+ SY0-401

## **Replay Attacks – CompTIA Security+ SY0-401: 3.2**

Even if the bad guys can't hack into your system, they may be able to temporarily morph into something that looks exactly like you. In this video, you'll learn how replay attacks can be used to gain inappropriate access to devices and how software developers can protect against replay attacks.

There is a lot of very important security information that is sent over a network. There's authentications that occur, user names are sent, passwords are sent. And a bad guy can take advantage of these with a replay attack.

To be able to perform this replay attack, the bad guy needs access to the network information. So they're either going to physically tap the network, so that they can receive a copy of everything that goes across. Maybe they sit in the middle by doing something called **ARP poisoning**. Or maybe they install malware on the victim's computer and simply have the victim's machine send the bad guy directly everything that was going across the network.

At this point, the bad guy will pick out the pieces of that information that will allow them to impersonate the original user. And then they'll replay that information across the network. Here's an example of how a replay attack might happen.

I have me, on my system. I'm communicating to a server and authenticating to that device, through a switch. And the bad guy has gained access to the switch and has told the switch to also send all the traffic down to the bad guy's laptop. Well, I'm of course going to authenticate to the server. So I'll send my packet through the network. And it will then complete its process to the server. But, of course, a copy of that has now been sent to the bad guy.

So he's captured that authentication process. That's probably going to be my user name sent in the clear and probably a hash password. You don't generally see passwords being sent in the clear across the network any longer.

Since the bad guy now has my user name and my hash password, he's going to attempt his own authentication request using that captured information. And now that that goes to the server and is captured by the server, the bad guy gains access. Because the server thinks that information came directly from me.

These days when you're using authentication like this across the network, we usually take advantage of a capability called **salting the password**. There's usually a session ID that's only in use for the duration of that session. Even if the bad guy was able to gather the session ID, the username, and the hash password, and use all of them together to form a valid authentication attempt, the server would know that that session ID was no longer available. And the entire process would fail. And the replay attempt would not be successful.

**Tags:** certification, comptia, replay attack, security

**Category:** CompTIA Security+ SY0-401

### **Spoofing – CompTIA Security+ SY0-401: 3.2**

The bad guys are very good at providing fake information to us. In this video, you'll learn how spoofing works and how pharming and phishing have become common ways to illicitly obtain our private information.

Spoofing is when you pretend to be something that you really aren't. Maybe you're a fake DNS server or a fake web server. We see spoofing in email, when we will get an email message that looks like it came from somebody we trust, but in reality that From address has been spoofed or modified. And it's not really who we think it came from.

You can also perform spoofing in a **man-in-the-middle attack**. You can change information as it's going by, so that the information that is received was very different than the information that was originally sent. You can even see spoofing happen with traditional telephones, with caller ID spoofing. So you look down when your phone rings. And it says there's a call from the White House but, of course, it's simply a spoof name and number that's popping up on your telephone.

One type of spoofing is called **DNS poisoning**. This is when we're changing the domain name server information itself. And you can change it inside of the server if you're taking advantage of a known vulnerability, although this is something that's a bit difficult to accomplish.

Another way to spoof DNS information is to spoof it on the client machine and not on the DNS server. If you change the host file on a client machine, it will use that host file before it ever talks to a DNS server. You can also do DNS poisoning by changing the responses that are sent back to the user. So as the user makes the request to the DNS server, we intercept that message going by and send back a fake response, pretending to be the DNS server. And when we do that, we can redirect the client to go to any IP address we'd like.

Here's how this might happen. We've got two users, User 1 and User 2. And we've got a DNS server and the bad guy sitting in the middle. The User 1 is going to want to query for Professor Messer.com. He needs an IP address of that particular web server. So he sends a DNS request on to the DNS server. In the DNS server, it has the correct address of 162.159.246.164. And it sends that response back to the client.

The client, upon getting that response, will simply fill in the gaps and then perform the normal communication to the professormesser.com server, using that correct IP address. The second user performs the same function. Sends a request out to the DNS server. But before he can, the bad guy sends an update message to the DNS server. And the DNS server does nothing to validate that. That means that the software in the DNS server is faulty. Once it receives that bad poisoned information from the bad guy, it changes the IP address for that particular website.

And now the second user sends the same request for the same domain name, but the response that goes back is the poisoned information that has been sent of 100.100.100.100. Now User 2 has received an incorrect IP address for the professormesser.com website. And now all the bad guy has to do is wait for that user to visit his malicious site.

We call this redirection to a bogus site, pharming, with a PH. This means that we were able to take advantage of a vulnerability in a client or a DNS server, to be able to redirect that traffic wherever we'd like it to go. Usually the bad guys are combining this pharming function with also phishing. So they'll send people to a site. And they think they're going to PayPal and putting in their credentials, but they're actually putting their credentials into

the bad guy's server. It's all a combination of redirecting the user and then presenting the user with something that looks familiar, but really is not.

This is very difficult for anti-virus and anti-malware software to stop, because everything looks legitimate. The DNS query performed properly. The IP address was received. The user went out to a site that looks absolutely correct. There's no malicious software involved from the client's machine. And there's no malicious software on the server. This means that we have to be extra diligent. Whenever we're communicating to a site and providing our private information, we need to check certificates and make sure that the site is truly legitimate.

**Tags:** [certification](#), [comptia](#), [pharming](#), [phishing](#), [poisoning](#), [security](#), [spoofing](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Spam – CompTIA Security+ SY0-401: 3.2**

Unsolicited emails are an ongoing security concern. In this video, you'll learn about spam, spim, and spit.

A type of attack that does not appear to be going anywhere quickly is spam. This is unsolicited emails we get. These emails are getting us or wanting us to click some links to buy something.

It could be sunglasses. It could be refinance your house. It could be pharmaceuticals. There are so many different things.

The problem with spam is that it is working for the bad guys. They're finding if they send a million messages, somebody will buy something. The problem, of course, is that it is so easy and inexpensive to send a million messages out that they're immediately getting some nice financial returns on these things.

And there is quite a different little network behind the scenes of botnets and spam senders that are creating problems for us. So it's an ongoing issue we have to think about when we're managing our email, managing our bandwidth, making sure we're keeping our systems very, very secure in our organizations. This has traditionally been used for advertising to get you to buy something, but the bad guys are also noticing that if they send a million messages with links that can then infect your machines those can also now become additional spam botnets.

That's how the bad guys are sending the spam these days. They're infecting your machine. Your machine now becomes part of a botnet. They send a command down to your machine saying, go send some spam for me. And your machine begins sending spam out everywhere.

Becomes a big issue, especially when those are all distributed throughout the world. And that's a pretty big challenge if you're trying to figure out how to stop these things. You can't go to a central source. It is a completely distributed process.

Another very common type of spam is spim. This is unsolicited instant messaging. This is when you pop open your IM and suddenly you're getting messages all over the place.

These bad guys have found if they sneak make some links into your instant message that you may want to click that. Because instant messaging is something that's a little more personal than spam and they think they can trick you into clicking those links. And those links can be really, really bad ones.

These are very, very directed and very, very specific types of spam. And generally, that's their only chance at this. It's something that also doesn't stick around. It's not something that's in your email inbox so they have to be very tricky and very specific.

And it's usually, of course, robots. There's no real people sending in these messages. These robots are trying to get you to click those links and have it send that information, have you infect your machine, have you log into a fake website, send your info out to somebody else. There's also something called **split**. This is spam over internet telephony.

It's kind of a bad name for it. Spit. This is the saliva of unsolicited messages. But because it is voice, it becomes very, very difficult to avoid this. You also can't filter out very easily with voice over IP.

Now one of the advantages of this is that voice over IP, like Skype, like your Google Voice, have made it very, very difficult for the bad guys to use this for bad purposes. So that's one advantage we have and one reason we aren't getting a lot of unsolicited internet telephony. But if you are ever in Skype, you know that Skype does have instant messaging capabilities so they're often piggybacking on some of the additional capabilities built into the internet telephony. So you may not be getting voice, but you may still be contacted through other means within those internet telephony applications.

There are a number of different philosophies to blocking spam, and many of these work in conjunction with another. One of the more obvious ways is to create a white list, which means the only email that comes into my inbox are things that had been checked off as real people. Sometimes this is an automated process.

The first time you send a message to somebody, you may get an automated response back saying, I don't know who you are. If you are a real person, please click this link, type this information in, and you will be added and your email will be sent on to my inbox. But the challenge, of course, is sometimes legitimate traffic might be blocked.

So we might want to approach it from the other direction. Have a blacklist. Stop everything that we definitely know is bad. Obviously, this can be a very, very, very large list, but this is something that third parties do keep track of and they have very, very big lists. This is one advantage of being on the internet is many people can all work together to create these blacklists.

The problem, of course, is if somebody legitimate gets on the blacklist, you may not be getting some legitimate traffic and that does occasionally occur. Somewhat of a smarter approach to resolve the spam issue is something called Bayesian filtering. This is where we're not looking at a whitelist or a blacklist and we're not looking for specific words in an email.

What we're doing is looking at the entire email. We're looking at words and phrases. And if we happen to see a number of words and phrases altogether at certain places within the email, you get a particular score.

And if that score is above or below a certain threshold, it either goes into your spam box or it goes into your inbox. So you have some adjustments you can do with this as well. Certainly, not perfect, but if you're trying to block the majority of the bad stuff coming in, Bayesian filtering may be a good way to do that.

These days, our spam filters are just built into our email clients, of course. We have sometimes whitelists, blacklists, and Bayesian filtering all built into what we're doing on our desktop. Maybe it's built into our organizations' Outlook Exchange front end.

Maybe we have a third party that does all of our spam filtering for us. They're almost always using a number of these different technologies. And in reality, you have to use all of these working together to really keep all of the noise and all of the spam out of your inbox.

**Tags:** [certification](#), [comptia](#), [security](#), [spam](#), [spim](#), [spit](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Phishing – CompTIA Security+ SY0-401: 3.2**

The bad guys know that the easiest way to steal your information is to have you type it in for them yourself. In this video, you'll learn how phishing happens and things you can do to protect against phishing and spear phishing.

**Phishing** is a very specific kind of social engineering that takes advantage of misdirection. It makes us appear that we are logging into our **Paypal account**, that we're logging into our bank account, that we're logging into our email account. But, in reality, we're logging into a page that has nothing to do with PayPal. It has nothing to do with our bank. And it has nothing to do with our email.

You'll usually get a message in your email that says, we've detected some unusual things happening on your PayPal account. You should log in right now. Click this link. And let's make sure you're adding in information and confirming the information that's in your account is correct. But the reality is, when you click that link, you're actually brought to a page that looks exactly like PayPal. But the reality is, it's on another site. I blurred it out, so you would not see what the site is. But you can see this is definitely not a Paypal.com address that you can see here. It does have the word Paypal.com in it, though.

So if you aren't familiar with this, you might be taken aback a little bit, initially, for seeing this message that pops up, warning you. But you may want to go right to this site and log in. And it says Paypal right there. So let's go log in and make sure everything is OK with our account. But don't be fooled by this. Usually the URL is going to tell you that this is definitely not Paypal. You should look for that Paypal.com or your website.com.

In fact, what you should do is never click a link inside of an email, never click a link inside of an instant message. If you get a message that says, we think there's a problem with your account, open up a browser window, type the name of your bank into that browser window, and go there directly. That way you can be sure you are going to the right site.

In fact, with this particular site, you'll notice that this looks very similar to Paypal. Everything is almost exactly the way it should be. But notice there is something that is a little bit off. You may find that certain images don't load. Or you may find that, say, there's a misspelling on the page. That's very, very common for some reason, to find misspellings in the email that's sent to you and to find misspellings in the page. This one does not have any misspellings, I don't think. This one is very, very well done. It looks just like PayPal.

More and more, though, this is done over the phone. People will call you and say, we understand there's a problem in your account. We've been looking at your purchases and your credit card. We think there are problems with that. Let's confirm this. Can you give me your credit card number? Thank you very much. Now, can you give me the last four digits of your social security number? Why, thank you so much. Now, tell me more about your family life and what your mother's maiden name might be.

So the vishers, we call them, which is phishing over voice, over the phone, have gotten very, very good at the social engineering aspect. And we trust the phone a lot more than we trust what's on the web sometimes. But don't be fooled. These people are really out to get your personal information.

Here's the same **Paypal page**, blown up, so you can really see it. And, boy, it looks really legitimate. It looks exactly like my **Paypal site**. There's nothing about this that would make me think that this is unusual, except for the URL that's up here. And that's really the thing you should be looking for, is to determine, is this really a legitimate site? Or is this something that's trying to fool me into typing this in? And that's the reason you'd never a click a link in an email and never click a link in instant messaging, even if it is a legitimate message from PayPal. I don't click those links. I make sure I go directly to the site, by opening my browser and typing Paypal.com right here on the top.

Traditional phishing just throws a net out there and tries to catch whatever it can. Sometimes you're catching a big fish, but more often than not, you're pulling up the tiny little fishes. You're pulling up an old tire or a rusty can. And if somebody was to get my Paypal information, they would find, this wasn't really worth my time.

So what the bad guys are doing is something called spear phishing. They're going after the really big gets. They're really focusing their efforts on getting a particular amount of information, particular logins to financial sites, or something that really, really interests them. And by doing this and getting a little bit of background information, they can make their emails more believable. They can add real-world information into them, add friend's names to the list, and be able to make you think that this really came from a friend, therefore, you should trust it.

If you were to spear fish the CEO, you'd call that whaling. You're going after a really big fish and trying to get something good out of that. Some examples of this, in April 2011, there was a company named Epsilon. They handle emails, sending out a lot of emails for third parties. They had less than 3,000 email addresses directly attacked inside of their organizations. So somebody knew the internal email addresses for Epsilon and started sending a bunch of emails internally. They really hit 100% of the operation staff. Because they knew if they got access to the operations logins, they would have access to the entire database of email addresses and, in fact, they absolutely did. And they were able to get millions ' of email addresses.

Now, initially, you think, big deal, they've got millions of email addresses. But, the reality is, they're now going to use those to send additional phishing attacks and try to get additional information from there. When they send a message into a particular group of people that had them click a link and that link downloaded anti-virus disabler. It loaded a key logger. It loaded a back door that got a remote administration tool on it. And then they started gathering a lot more information.

It seemed like a legitimate login page. They logged in and, ultimately, they were infected. But what if it isn't emails? In also April 2011, **Oak Ridge National Laboratory** was hit as well with a phishing attack, a spear phishing attack. It was an email that was sent. And the From came from the **Human Resources Department**. Boy, if I saw an email from the Human Resources Department, I'd want to open that up. And it said, you need to log into your HR account. You need to make sure that your benefits, or something of that sort, were available. You can send all kinds of interesting things in an email from the HR department. It targeted only 530 employees. And there were 57 people that clicked on the link. There's probably another story here about why all of these other people didn't read the emails from the HR department or didn't feel that they needed to click, maybe they were trained very well, not to click links inside of their email, but 57 people did.

And, of those 57 people, two machines were not updated with malware protection. They were infected. They went to the site. They logged in and immediately were infected. Data was downloaded. Servers were infected with malware from there. This was a big problem, because the **Oak Ridge National Laboratory** performs research on nuclear testing and other types of research for the federal government. So once you get those two machines infected inside that organization, now it can start hopping around and doing other things, because the security on the inside of the network, generally, is much more open than the security from the outside.

So you get that malware inside the organization and, in this particular case, information was stolen. This becomes a big issue for organizations that want to be sure none of their internal information gets out. And, from the point of a spear phisher, they're going to focus on those particular users, make it very, very believable, in the efforts that they can get inside, get infected on those machines. And now they have access to a lot more information.

### **Vishing – CompTIA Security+ SY0-401: 3.2**

Why would the bad guys hack into your computer when they could just give you a call? In this video, you'll learn about vishing (voice phishing), and how the bad guys can even fool you into calling them yourself.

The bad guys will do anything to extract our personal information from us. They'll get it from us on web pages. They'll try to use email. And now, of course, they're trying to use the telephone.

And this process of using the phone or your voice to try to gather personal information is called vishing. Taken after that email term of phishing, but the V in vishing is for voice. This provides a direct connection to you. You can hear a voice on the other end of the phone. It's a real person, and they're telling you all about the problems that have occurred on your computer.

They may say they're calling from Microsoft and they need to remote access into your computer to see if they can solve problems that they happen to be seeing on their side. But of course, this person is not for Microsoft, but they've used the telephone as that initial conduit to you, and now they'll manipulate you to be able to get to your personal information.

Having this voice connection adds a level of trust. When you receive an email or you look at something on a web page there's no personal connection. But the bad guys know if they can communicate to you with a voice connection that the trust level will certainly go up. And so many more people have a phone than have email or have a web page that they would browse to. There is so much potential for the bad guys to connect to you over a telephone than any other method.

In an interesting turn, the bad guys are having you call them. They may send you an email that says that your very important financial account is locked, or they got your message about the utilities that will be disconnected tomorrow and you can call this number to confirm, or your cable television or your internet connection may be moving today and you need to call if there are any problems with that. So they may leave a phone number for you on that page or in that email, and that may prompt you to call them thinking that you're calling your financial company, you're calling your cable company. But in reality, you're really calling the bad guys.

You could also be fooled by very professional front end. The bad guys may create an **IVR**, an **Interactive Voice Response unit**. One of those that when you call says, welcome to First National Bank. Press 1 for a teller. Press 2 for a vice president. Press 3 to contact support.

Those types of interactive voice responses make things sound very professional. And they're very, very simple to set up. So why wouldn't the bad guys put that at the very first thing you get to, which, again, adds to your level of trust about who you've contacted.

If the bad guys are going to call you, they're going to also change the caller ID information that pops up on your phone. If the phone rings with an unusual number, but the name says Microsoft then you may be a little more trustworthy of somebody who's on the other end of the phone that says they are indeed from Microsoft, and they need to remotely connect to your computer. It's a very trivial thing to be able to manipulate that caller ID information, and the bad guys absolutely take advantage of that.

Of course, why would we want to call someone on the phone or send an email when we can simply send a text message? We've become so comfortable at receiving text messages, not only from people we know, but from the organizations that we do business with that it might look absolutely legitimate to receive a text message saying that you are a financial company and you need to reply with certain account information.

So you need to be careful about **SMiShing**, which is the **SMS** version of phishing so that you can avoid having your personal information get into the hands of the bad guys.

**Tags:** [certification](#), [comptia](#), [phishing](#), [security](#), [vishing](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Christmas Tree Attack CompTIA Security+ SY0-401: 3.2**

By changing a few bits inside of a network packet, you can cause a number of things to occur. In this video, you'll learn about Xmas tree attacks and you'll see what happens when I run a Christmas tree attack against my own router.

A **Christmas Tree Attack** is a very well known attack that is designed to send a very specifically crafted **TCP** packet to a device on the network. This crafting of the packet is one that turns on a bunch of flags. There is some space set up in the TCP header, called **flags**. And these flags all are turned on or turned off, depending on what the packet is doing.

In the case of a Christmas tree attack, we're turning on the Urgent, the Push, and the Fin flags. And you can see, here's an example of a screenshot of Wireshark, where Urgent is set. The Fin is set. And Push is set. So we've got these three different bits that are set in here.

Now, normally, we wouldn't see some of these combinations of bits being turned on or turned off inside of the packet. So it's very unusual. And having so many of them there and having these 1s and 0s there, mean that this particular section of the flags of a TCP packet are lit up like a Christmas tree. And that's where the name comes from.

It's an odd combination. The interesting part is how do the remote devices respond when you send a Christmas tree attack to them? And how they respond might give you an idea about what's on the other side of this. Different devices respond to different ways. So this may be a very good way to get some reconnaissance from a device.

Sometimes the device has no idea what to do. And you can slow down the device, because it has to really look through this and understand what's going on. In my particular case, I ran this attack on my internal network as I was building this presentation. I ran it against the router that goes out to the internet. And, about two minutes later, the router rebooted. And I realized, that's odd. That doesn't happen with my router. My router is very, very reliable.

So I ran the Christmas tree attack again. Two minutes later, the router rebooted. I realized I had a router that was susceptible to this Christmas tree attack. And, granted, my router is a very, very old router. It's one that was not doing the latest wireless technologies. It's one that was, in fact, many, many, many years old. And I realized, at that point, it's time to upgrade the router. I don't expect anyone in my house to be doing a Christmas tree attack. But it spoke to the reliability of the router. So I upgraded the piece of hardware, so that it would not be susceptible to this attack.

We're going to run an attack again in this video and see if we have the same problem. This is something that's very, very easy to see, if you have an intrusion prevention system. They have signatures that are specifically designed to identify Christmas tree attacks when they're going through your network. And if you have a packet captured device that we're going to run here, you can go through there and look at the packets themselves to see if there is this combination of bits turned on and turned off right inside the packet.

I'm going to use two tools to be able to show this attack, when it's occurring, and to perform the attack itself. And, as a security professional, they are probably tools that you have already. The first one is **Nmap** which you can download from [nmap.org](http://nmap.org) or

insecure.org. They go to the same place. That will be the scanning tool that I use to perform the Christmas tree scan, the Christmas tree attack against this router that I have in my environment.

We're also going to use Wireshark. Wireshark is a packet capture protocol analysis device. We're going to capture packets in real time. And I'm going to see what the results are after attack is over. So before we get started, let's pull up our capture options. I'm going to go from my ethernet port. And I'm just going to start up the packet captures on my network. And it's going to start sending the packets back and forth behind the scenes. You can see that those are going.

And, in the meantime, I'm going to perform this Christmas tree scan, since I'm not running as root on my computer. I'm going to do a [INAUDIBLE], so I can run this Nmap scan as root. I may not have to do it for Christmas tree scan, but it's become a standard thing that I do when I run an Nmap scan. The flags to perform a Christmas tree scan is the flag-s and a capital X. And that's the Christmas tree. That's the scan for Christmas tree.

And I'm going to do it to 192.168.1.1, which is my router. When I hit Enter, and it's going to ask me for my password, since I'm asking to run as root for this, and hit Enter. It's going to perform the Christmas tree scan. And then it's done. And it performed a lot of information and found a number of closed ports on this device. Some of them were open and filtered. Christmas tree can't tell you if it was really open. But it knows that it did not get a response when it heard these particular ports come back.

Let's look at our packet capture and see if we can see the exact flags that were inside the **TCP** packet when the scan ran.

I've now stopped my packet capture. I found a frame that is part of the Christmas tree scan. But we're scanning, we're sending these Christmas tree packets out to the router. And we're sending a few thousand different scans. They're all on different port numbers. So you'll see it all coming from one port number. And it's mixing up and randomizing the set of ports that it's sending back and forth.

So we'll just choose one of those. Let's get rid of the hex decode down at the bottom, make it a little bit smaller. And let's have a look at the scanning part itself, the **TCP flags** that we have here. We can see that we have a Reserve not set, Echo not set. Urgent is set. Push is set. Fin is set. So there are the flags— I'm going to move up a little bit so you can see where these flags are— this is in the frame going across the network. It's an ethernet frame with the Mac addresses of these two devices. It is an IP frame. You can see the IP addresses associated with this scan. And this is the TCP part of the frame. And in the TCP, you have, of course, source and destination ports.

And then, down here a little bit further, are all of the flags. And that's where I'm seeing the flags associated with the Christmas tree scan. Quite a lot of information here. Obviously, if you were doing this by hand, by eye, with a human being, this could be very, very difficult to identify. It's practically impossible to see this, with all of the other normal traffic going through your network. Which is why it's so important that you have these intrusion prevention systems on both your network and your host, so if somebody is attacking a single machine and using some of these well-known methods, you'll be able to see it alarm immediately.

Also, as a followup, I was able to keep a ping going to my router the entire time that I've been doing this video. So now I can be assured that a Christmas tree scan is not going to bring down my brand new router. That's something you need to keep in mind. If somebody is performing these scans on your network and they're causing systems to go offline or causing a denial of service situation, then you may need to get updated firmware. You may need to get updated hardware. Or find out what you can do to prevent somebody in your network from causing a denial of service to your very, very important systems.

## **Privilege Escalation – CompTIA Security+ SY0-401: 3.2**

If a bad guy can escalate the privileges of a regular user, he'll have greater access to the system. In this video, you'll learn about privilege escalation and how to mitigate privilege escalation.

Privilege escalation is when you're able to gain a higher level of privilege on a system even though you're not supposed to have that privilege. This could be due to a vulnerability that exists on the system or it may be a flaw in the operating system you happen to be using. Generally speaking, having a higher level access means that you have more capabilities on that server or that system.

Generally, when we do privilege escalation we're trying to get the highest possible privilege. So if you're on a Windows machine, you want that administrator access. Or if you're on a Linux machine, you want that root access. And that's obviously a big problem because if somebody's able to get that level of access on a system, they can do anything to that operating system they'd like.

You'll notice when Microsoft sends out every month their security patches, if any of those patches are identified as allowing a privilege escalation those patches are usually set to a very high priority and people like to be able to patch those as quickly as possible. As soon as the bad guys know that there's an opportunity for a privilege escalation they're going to want to try to take advantage of that. So you want to prevent that from occurring by keeping your systems patched immediately.

If the bad guys know that there is now a new vulnerability that allows them to have a privilege escalation they will take advantage of it. So you want to deploy those patches as quickly as possible. There's also something called horizontal privilege escalation. That means that one user who has normally access to just their files might also be able to gain access to another user's files, but not necessarily have any additional access to the overall system.

Since many privilege escalations occur because of a known vulnerability, it's important to patch as quickly as possible and that might resolve a number of privilege escalations on your servers. Your antivirus and anti-malware software might also be able to block this privilege escalation, especially if it's a known vulnerability and there's a known executable that takes advantage of that vulnerability.

Many modern operating systems use a technology called data execution prevention. This keeps the executable running only in areas where it's allowed and prevents it from going outside that area and allowing a privilege escalation. Another operating system feature is called address space randomization where the data is put in many different places and it's randomized every time. This prevents malicious software from being able to take advantage of a buffer overflow because there's not a known memory address where certain data might always be located.

**Tags:** certification, comptia, escalation, privilege, security

**Category:** CompTIA Security+ SY0-401

### **Insider Threats – CompTIA Security+ SY0-401: 3.2**

A huge security risk is on the inside of your network. In this video, you'll learn about insider threats and what you can do to help secure yourself from this very intimate security concern.

We spend a lot of time and money protecting our network from people who are on the outside. But of course, we need to also think about protecting the resources that we have on the inside of our network from the people who are also on the inside of our network. If you look at the statistics for insider threats, they are a significant part of our security strategy. We give people all kinds of access on the inside of our network. That, of course, is why we have this concept of least privilege.

We want to give rights and permissions to people, but only just enough rights and permissions to allow them to do their job. We don't want to give everybody in the network administrator access to every device on the network because we want to be sure that we're minimizing all of the security threats for everyone on the inside, as well as on the outside of our network.

If you are inside the building of an organization, you automatically have more access than someone who's on the outside. In many organizations, there are specific rules and policies that deal with visitors who are coming into the building. In some organizations, a visitor can't even enter the room unless everybody turns off the screens of their computers, they put away all of their papers, and allow the visitor to walk through the room. In those very high security environments, it really requires some additional security to guard against those insider threats.

Having one of these insiders cause a security problem can also cause other issues with your organization. For instance, if an insider was to cause a security problem that allowed confidential information to get out, that means that people may trust you a little bit less. This might harm your reputation because if you can't protect from the people who are already inside the building and your employees then how can we trust you as an organization?

This could also, of course, cause outages and downtime. And of course, for a critical system to be down it could be costing the organization a large amount of money. And if your information that is proprietary gets out, you may be giving away the secrets to what you're doing as a company. So by guarding against these insider threats, you may also be ensuring the future of your organization.

Every year, Carnegie Mellon does a survey from the **Computer Emergency Response Team or CERT**. And this survey in 2014 is the US State of Cybercrime Survey. You can find out all about this at [cert.org](http://cert.org). They give interesting statistics, not only for threats from the outside, but also for insider threats.

And in this latest survey, 28% of the attacks that people had come from inside of the organization. So over a quarter of these attacks were insiders that were causing these problems. 32% of the respondents said that the damage from an insider attack was more damaging than someone from the outside. Having that level of access on the inside of your network certainly gives people more rights and permissions. And when there's a problem, they are sometimes creating more of a security problem.

In the 2014 survey, 75% of the respondents said that the insider incidents were never handled with any type of legal action. They were able to take care of things internally within the organization. So you have to think that just because you're not hearing about the insider threats doesn't mean that they're occurring.

In fact, in the vast majority of cases, you will never hear about these insider threats because they don't go to court. They're not made public. There's not a press release. They are simply handled internally.

**Tags:** [certification](#), [comptia](#), [insider](#), [security](#), [threat](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Transitive and Client-side Attacks – CompTIA Security+ SY0-401: 3.2**

If the bad guys can't attack a server directly, then they'll try going through a trusted neighbor. In this video, you'll learn how transitive attacks and attacks against the clients have become significant security concerns.

Transitive attacks are attacks that become very, very difficult to prevent. You have to spend a lot of time looking at configurations and making sure your systems in your network is not set up to allow something like this to occur. A transitive attack is where a machine A trusts machine B and machine B trusts machine C. Therefore, I can attack machine C and machine C will automatically be trusted by A.

Now this may not actually be something you want to occur. This may be something that's occurring just because the series of trust that's been set up. Maybe in reality you really did not want a trusting C, but because of this transitive nature of trust in operating systems it's something that may be there already. So again, it's something you really have to look for.

In network security, this is an ongoing concern. It used to be in older Unix systems, this was a normal part of the operating system. We would set up specific configurations in the operating system that allowed trust to many, many different computers. It skipped over step of having to authenticate every time we went to a server that was trusted.

But in those days, we didn't have to worry so much about somebody taking over our machines and then gaining access to everything in our organization. These days, our systems are designed not to allow those trusts by default. In fact, these days it's very, very common for a machine just not to trust anybody.

Our firewalls don't trust anybody. Our computer operating systems don't trust anybody. Our server operating systems don't trust anybody. And in reality, with the type of security concerns we have today it's probably a good idea to keep things running that way.

Firewalls are often used to help with this as well, especially if you have many different business partners connecting into your network. It's very common, unfortunately, for people to set up a firewall and forget to block access to different business partners. These days, we lock them down pretty well.

Obviously, your firewall isn't going to be able to block everything. So if you have business partners coming in in your organization, some people may set up completely different firewalls and completely different kinds of access to all of those. But ultimately, you've got a hole there so that your end users can access that business partner or the business partner can come into your organization.

And unfortunately, when you build a hole and build access into a firewall, you can have a lot more than just what you designed to come through that firewall hitting your network. So these transitive attacks are things you must keep in mind whenever you're setting up connections between systems and between organizations. Of course, the bad guys have noticed that we're putting firewalls, we're hardening our servers, we're making it very, very difficult for people to go directly to the source to get the type of access they want.

If a bad guy wants to get to a database, it's very, very hard now to go to that database server. We put all kinds of security methods in place to prevent direct access to that

database. So the bad guys have decided, well, if you're going to protect the server, let's go now and attack the client because the client is not going to be protected from the server. Your client has to talk to the server. Therefore, that's a great place to go.

And if you can find an application that is badly programmed that will allow me access to that data in a way that perhaps you were not expecting then I now have access to everything in the database. And that becomes a huge concern. So our bad guys are now hitting those clients.

And we have so many different applications running on our computers. We've got browsers. We have media players. We have email applications. Each one of those may have vulnerabilities associated with them that would allow the bad guys access to your computer, access to your data, or access to the server by hopping through that application.

And one single tiny bug in any one of those applications can now cause a huge amount of data to be exploited on your servers. So that's why you have to make sure that you're keeping your operating system updated, you're keeping your applications updated. You want to avoid that single vulnerability so by staying up to date with all of these patches and all of these updates, you can be at least a little more sure that your end users are going to be protected from somebody trying to take advantage of some of these client side attacks.

**Tags:** certification, client-side, comptia, security, transitive

**Category:** CompTIA Security+ SY0-401

### **Password Attacks – CompTIA Security+ SY0-401: 3.2**

The bad guys don't need to know your password; they'll figure it out themselves. In this video, you'll learn the techniques that the bad guys use to reverse-engineer your password.

To gain access to an account or you're authenticating, you need to provide your username and your password. The username is usually something that is easily seen. Something you can see on the screen. It's often sent via plain text over the network, which means it's your password that is really the most important piece.

It's something that is not sent in the clear over the network. It's often hashed, which means you have no way to reverse engineer what that original password happened to be. So the idea that the bad guys have is they'll try every possible scenario to try to determine what your password might be. This is called a brute force attack.

They will very methodically go through every possible combination of lowercase letters, uppercase letters, numbers, and special characters to try to determine what your password happens to be. And if it sounds like this could be a very involved and very long process then you would be correct. This takes a lot of time to be able to try every possible combination.

If you've ever forgotten your own password to a corporate account or an online account, one of the things you'll notice is after a certain number of attempts it locks the account completely. And at that point, you have to receive an email or call somebody to unlock the account and reset the password. And that's to prevent these brute force attacks.

The bad guys know that this process of over and over typing in a username and a password is not only slow, but as you can see, your account will quickly become locked out. That's why the bad guys gain access to the servers and gather the hashes directly from the authentication database itself. So they'll get a list of all the user names and all of the hashes associated with those usernames. And the hashes are the hashed password that was sent into that system.

And what they will do is begin their own calculations. Well, they'll take their random guess. They will hash it. And then they'll compare that hash to what was in the database. And they'll do this over and over and over.

Because they have the hashes, they don't have to worry about locking down an account. They can keep trying this as much as they'd like. Brute forcing the hash has its own disadvantages, of course, because the bad guys have to physically calculate what that hash is. And that requires a number of CPU cycles.

If you're simply typing in a username and a password and trying to authenticate, there's not a lot of **CPU cycles** involved there. But if you're trying a million different calculations, you're trying a million different password iterations for a user, there's a number of CPU cycles that will be used there. So it's not a cakewalk for the bad guys. They still have to do some significant work to be able to compare their calculated hash with what the stored hash might be.

Different applications and different operating systems will store this authentication hash in different ways. It's pretty standard across the application or across the operating system, but you can't take a hash from a Linux device and compare it to a hash from a **Mac OS X or a Windows device** because those operating systems may store the hashes in different formats. This is an example of hashes that I grabbed from my Windows 7 device.

So I've got these usernames. Jumper Bay, Carter, Jackson, O'Neill, and Teal'c. Those are on my Windows 7 machine. There are some IDs associated with those usernames.

And everything there at the end— that long string of hexadecimal numbers and letters there— that is the hash that is created by Windows 7. So if somebody grabbed this database, they could then take that information and begin their process of brute forcing those hashes.

If you're trying to determine someone's password, you don't necessarily have to go through every possible iteration of lowercase numbers and uppercase numbers and letters and special characters. It might be a little faster if you simply chose words out of a dictionary. That tends to be what we use for passwords. It's something that we can remember. It's a word or a phrase that makes sense to us.

So a dictionary is the perfect place to go. And we can even make it easier by starting with a list of passwords that are most commonly used. Words like password is one of the most common passwords and probably not a good idea to use for your authentication. But there's other words, like ninja and football that are just as familiar and just as commonly used.

There are large databases on the internet. You can download from many different places that have a list of the most popular passwords or a list of every word that happens to be in the dictionary. And the bad guys are going to use that list first before they go into randomizing and trying to determine what you might want to find from special characters or upper case.

For the smarter people, they're going to use those more complex passwords. But a dictionary attack is definitely going to grab the folks that are trying to use a simple password to remember, which unfortunately, is a very simple password for the bad guys to guess as well. Of course, not everybody uses a very common name as their password. Most people are going to add special characters or perhaps letters or numbers to replace other letters or numbers in a word.

And one way that the bad guys can use to find these passwords is something called a hybrid attack where they're going to use a very common set of dictionary words, but they're going to change them up just a little bit and try different variations of those words.

So it's not uncommon to see somebody that might use the password apple, but they'll put the number 1 after the apple. Or they might use ninja and put the number 9. Or they might change a T to a 7 within those words.

And the bad guys recognize that you're going to do this so there are options in their brute forcing programs where they can use a dictionary, but change it and add numbers to the end, replace the letter l with the number 1 to see if perhaps you're doing the same thing with your password. This is, obviously, going to take a little bit more time. You have to try one dictionary word and then many different iterations of that dictionary word, but obviously, it's going to be a lot easier than going through every possible iteration of every character and every number and every special character as well.

This is something the bad guys can easily configure inside of their password cracking or brute force software so don't think that simply changing a number or changing a single letter is going to protect you because the bad guys have already thought of that. There are a number of cryptographic theories the bad guys can use to help determine what your password might be or to be able to duplicate a hash. One of these is called the birthday attack.

And the way that this works is best described by this. In a classroom of 23 students, what is the chance of two of those students sharing a birthday? Now if you're thinking about this off hand, you may think it's 1 in 365, but the example we're using here is every student is comparing their birthday to every other student. So the reality is if you get 23 people in the room, there's about a 50% chance that one of those students is going to share a birthday.

In the world of cryptography, this is called a hash collision. This is something that really should not be happening. But unfortunately, a number of the hash algorithms that we use have the potential for colliding.

A hash collision is when you have one type of plain text that creates a hash, you have a completely different plain text that creates the same hash as well. This isn't something that's supposed to happen. We design our hash algorithms so that we don't have these types of collisions. But unfortunately, these do exist with certain types of hashes.

So this is a great thing to find for the bad guys because they may not necessarily need your original text, and they might be able to use a different kind of text, but still have exactly the same hash at the end of the day. This is something that the attackers will then use to create multiple versions of this plain text, especially if they can only slightly modify some plain text and then have exactly the same hash on the other side.

They might be able to make changes to a document being sent across the network. On the other end, you can look at the digital signature of that hash and it matches what was sent, but the reality is that the bad guys changed something between one side and the other. And of course, this can be used for much more than just digital signatures. It could be used for the certificates that are used to encrypt data on a web server. So you might think you're going to a trusted web server, but the bad guys have changed and found a collision hash that allows them to build a certificate that looks like yours, but it's really owned by the bad guys.

One thing that can help prevent these hash collisions is to use very large hashes. The larger the hash, the more difficult it will be to find a collision, and the safer that plain text is going to be. We talked earlier about brute force attacks with hashes where the bad guy would try to guess what your password was, they would calculate a hash, and then compare that hash to what was stored.

What if all of the hashes, though, were already precalculated and all of those hashes were not only precalculated but also optimized to be able to store and find them very quickly? This type of database is called a rainbow table, and it's used quite often to try to reverse

engineer those hashes. This could be a very, very fast way to determine what a password is, especially when you're dealing with larger and larger numbers of characters in a password because those take a lot longer to go through and try every possible iteration.

It's very easy to go through set a five character passwords and try every possible combination, but what if it was a 12 character password? That takes a lot more time. Well, if you've already done the calculations and stored them in a rainbow table, now you can very, very quickly simply search through the table for exactly what you're looking for.

Although having these rainbow tables can greatly speed up the process, you still need a separate table, a separate database of these rainbow tables for each type of technology. That's because Windows 7 uses one way to hash and MySQL might use a completely different way to hash these passwords. That means you need to build two completely separate sets of rainbow tables.

So this hard work has to be done well before you go through the process of trying to reverse engineer. And if there are many different technologies that you need to reverse engineer, you will need a separate rainbow table for each one of those. Since we recognize that the bad guys can easily reverse engineer these hashes with a rainbow table, one thing that our application developers are doing is salting the passwords. They're adding our password and so random bit of information and storing that information on the server. That way even if you are able to obtain my username and my salted hash, you would not be able to reverse engineer this with something like a rainbow table.

**Tags:** certification, comptia, hash, password, security

**Category:** CompTIA Security+ SY0-401

## **URL Hijacking – CompTIA Security+ SY0-401: 3.2**

One way to redirect your browsing activity is to force you to a site that you weren't intending to visit. In this video, you'll learn the techniques used to hijack URLs.

A **URL hijack** is when you think you're going to one website and you end up going to a completely different one. And the URLs may look very similar or they actually might be very different. This hijacking can take place through a number of different mechanisms that we'll talk about in a moment.

The reason these hijacks take place is primarily money. If they can redirect your eyeballs to a site they own then the bad guys can make money off of a mistake that you happened to make when you were typing in a URL or when your URL was redirected to their website. There is a questionable market for badly spelled domain names.

These badly spelled domain names may end up gathering a lot of people to a site. And the owner of the badly spelled domain can go to the actual spelled domain owner and ask them if they would like to buy that domain since a lot of their legitimate users are ending up on this third party site. Wouldn't it be better, Mr. Business Owner, if they ended up on your site to begin with and all you'd have to do is pay me money for that particularly badly misspelled domain name?

Sometimes these badly spelled domain names can be used to redirect people from one particular website to a competitor's website. This, obviously, has a number of legal issues associated with it. When this has occurred in the past there has been a lot of legal courtroom work being done. And it usually ends up with that domain name being corrected so that it does not point to a competitor. But obviously, this is not something that is good for anybody while this redirection is going on.

Sometimes it's done for a phishing. Somebody wants your username and password. They're going to redirect you to a site that looks very much like your bank's website. Or it looks just like PayPal's site.

And you put in your username and password, and now they have your login credentials. And they'll go to your real bank account or your real Paypal account, and then they'll move money into their account. So this may not be something that's just an annoyance or somebody trying to make advertising money. This could be someone trying to get to your personal data, your personal information, or your bank accounts.

Sometimes you become infected with malware that will take your legitimate URLs and redirect you to a different website. We often refer to this as browser hijacking, but it's the same type of idea where they'll take a legitimate URL and either put their ads around it or redirect you to a completely different side altogether.

This type of URL hijacking that takes advantage of a badly spelled name is often called typosquatting or brandjacking. And it doesn't have to be a very obvious misspelling. It could be something that's very minor in the misspelling. And it may be something that a lot of people do.

For instance, `professormesser.com` versus `professermesser.com`. At first glance, they even look identical on the screen, but you'll notice professor in this case spelled with an o. Professor in this case spelled with an e. And obviously, you can find somebody who might misspell that when they're typing it in and they'll end up on the wrong website.

Maybe it's somebody who's just trying to type the name in legitimately, but they make a spelling error, like `professormeser.com`. This is where they miss an S somewhere in there, and they end up on a third party website instead of the site they were originally planning

to go. Or maybe the bad guys would just like to use a completely different phrase. Maybe you think my website is professors-messers.com. so they'll add an S to the end or they'll try to find a derivative of the name and also have that go to a third party site.

Then maybe a case where they use exactly the same name, but they use instead a different top level domain name. You see this most often if somebody owns a .org or a .net. The bad guys will get the .com version of that since that tends to be more popular. And instead of going to the site you think you're going to, you'll end up on the bad guy's third party site.

These URL hijacks should be something we're always looking out for, especially if we're planning to add private information or financial information to a website.

**Tags:** [certification](#), [comptia](#), [hijack](#), [security](#), [url](#)

**Category:** [CompTIA Security+ SY0-401](#)

**Watering Hole Attack – CompTIA Security+ SY0-401: 3.2** The most successful attacks happen when the target is least expecting them. In this video, you'll learn how watering hole attacks can be a very effective way to gain access to an unsuspecting target.

If you're really diligent with your network security then it's going to be very, very difficult for the bad guys to find their way into your network. You may have a very secure firewall. You may follow all the best practices for security. You're not even plugging in those random USB keys that you might find in the parking lot.

Well, this creates a problem for the bad guys because they want access to your data. You're not responding to phishing emails. You're not opening email attachments and running them on your Windows desktop. So you're really preventing those easy ways for the bad guys to get into your network.

So instead of the bad guys trying to get in, what they're going to do instead is wait for you to come to them. So one of the things they're going to try to find out is where you go when you leave the building or when you access other sites outside of your private internal networks. This is your watering hole. This is where people go from the inside of your network to have lunch or to a popular website that people like to visit.

This is going to, obviously, require a bit of research to try to determine what sites the people within your organization go to. If they can somehow make you come to the watering hole, they might be able to take advantage of you there. Here's just one example of what a watering hole attack might look like.

Maybe in your organization there is a popular coffee shop or popular sandwich shop just around the corner, and a lot of people from your company will go to that sandwich workshop and put in their orders. Perhaps, go just before lunch. Go into the web browser. Go to the coffee shop or the sandwich shop URL. And then put in your entire order so that someone can then go by and pick up that food.

Something that happens all the time when we're working, and it's something that may also be very easy for the bad guys to take advantage of. They're now going to go not to your website, but they are going to go to the sandwich shop website and they're going to try to find a way in there. So they'll see if there's vulnerabilities that allow them access to the sandwich shop or the coffee shop website. And from there, they may be able to put information in where they could capture this and then infect your machines on the other side.

Or maybe from there, they're able to send you emails and you'll trust an email from the sandwich shop a lot more than you might trust an email from somewhere else. So you

may end up clicking one of those attachments inadvertently and, of course, that would then infect your computer. Obviously, in those particular cases you may be infecting everybody who visits the sandwich shop or the coffee shop, but the bad guys don't care. They're really ultimately trying to get to you.

And even if they infect everyone else, as long as they can get that one person to get infected inside of your organization that may be just what they need to then get on the inside and then spread out and gather whatever information they might need. So you have to be very careful of not only your internal network, but also be very careful when you're going to the watering hole.

**Tags:** [attack](#), [certification](#), [comptia](#), [security](#), [watering hole](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Shoulder Surfing – CompTIA Security+ SY0-401: 3.3**

What could be easier to social engineer than a shoulder surf? In this video, you'll learn about shoulder surfing and some methods to protect yourself against this tactic.

From a social engineering perspective, shoulder surfing is exceptionally easy to do. And the reason people want to shoulder surf and see what you're doing on your laptop or your tablet computer is because you have access to very important information. You may have sales information in a spreadsheet. You may be looking at internal company details.

And there is always somebody else who's out there, a competitor or somebody who can sell this to a competitor who would like to learn more about what you are doing. This is very, very simple. And you could do it in so many different places.

You may be in a coffee shop. You may be at an airport. You may be on a flight. You may be somewhere where your machine is just open and available.

I've seen people get up from a coffee shop and walk and grab a coffee, look at something else inside of the coffee shop, and leave their computer unattended with the information right there on their screen. They were working on a spreadsheet, they were working on sales information. They don't know who I am. I may be a competitor of theirs and yet, they left that machine now unattended.

There you can also surf from afar. There are people that will get a room in a building next door. They'll get high-powered binoculars. They'll get some telescopes. And they'll watch and see what's going on.

You can also watch from afar by monitoring webcams and setting up other views of what you may want to see. Whether you're setting up a webcam that can spy on a desktop or whether you're setting up a webcam that can spy on what other people are doing inside of an organization, all of these details become extremely important to consider when you're trying to stop somebody from doing shoulder surfing.

So how do you prevent this? How can you keep somebody from seeing what's going on? Well, one way is just to be aware of what you're doing and where you are.

If you're in an environment where other people can see your computer, perhaps, this is not the best time to bring up that confidential spreadsheet or that confidential PowerPoint presentation. You also want to think about maybe adding some additional hardware to your computer and doing things, like these privacy filters.

Probably the biggest manufacturer of these is 3M, and they work remarkably well. I was on a flight, literally, sitting next to somebody. I'm in coach so I'm really right next to somebody. And I'm watching him work on his computer.

And I'm thinking he's a little bit crazy because I see his computer screen and it's completely black. I could not see a thing of what he's doing on his computer and I'm sitting right next to him. So these filters are designed so that you can only see if you're looking straight on at what's on the screen.

If it turns in any way, it now blacks out and people who are sitting next you have no idea what you're doing on your screen. It is a very, very nice way to protect yourself if you're in those types of environments.

You should also make sure that your desk is not one where your monitor is visible. Make sure you're not working in a place where people are walking by and can see what you're doing. This is a very big concern for organizations where you're dealing with financial information and medical information. That information should always stay private.

And perhaps, this is the best thing I can tell you is just don't sit in front of me on a flight. It's so hard not to see what somebody's doing on their screen. It's difficult not to see what's happening there. It's right in front of you.

Maybe you should just pull up you're Angry Birds, pull up solitaire and do something else if you know that you're going to be on a flight and other people are going to be around you shoulder surfing.

**Tags:** certification, comptia, security, shoulder surf

**Category:** CompTIA Security+ SY0-401

### **Dumpster Diving – CompTIA Security+ SY0-401: 3.3**

One man's trash is another man's security exploit. In this video, you'll learn about the dangers involved when disposing of your organization's rubbish.

This can be a very, very messy social engineering exercise, but it's what the bad guys are doing to get information about what's happening inside of your organization. The term dumpster diving came from this brand name of Dumpster here in the United States. This Dempsey Dumpster. It is, as I'm told, very similar to a rubbish skip in the UK and elsewhere in the world.

It's a garbage bin and it's somewhere where people are throwing their trash out. And it's placed— usually it's a mobile device in a larger organization. These are around the back of the building. And every week or so a truck shows up, takes away the old and leaves an empty garbage bin in its place. And in the meantime, this becomes a place where people are putting a lot of interesting information.

They're throwing away things they should absolutely not be throwing away. Internal company documents, information about names and email addresses inside of the organization, or really, really important data that might be private. People, unfortunately, are not following the right procedures and throwing out information they absolutely shouldn't be.

Sometimes even make it so easy for me. They'll put it in a bag. They'll put it in the garbage.

All you have to do is show up with a truck, pull out your truck, throw the bag in the back of the truck, and you're off. And you can take those bags somewhere else and open them up later and see what's inside of those. You can get a lot of interesting details here. So it

could be phone numbers that you could then use as social engineering to call somebody directly.

Hi, Mary. It's Bill in technical support. And you already know Mary's name. You already know Mary's number. You're just a little bit farther along.

Even better, you can call someone else and say that you're somebody internal in the organization because you have names, email addresses, and phone numbers. The timing is very important for dumpster diving. Usually, it's once a week or once a month. You can also get very interesting information, depending on what time of the month it might be.

You may be finding that you're at the end of a month or end of a quarter where a lot of information is thrown out right after that. They're purging their archives and freeing up room. They may be throwing out some very, very valuable information. So you may want to find out when their schedule is corporately and find out when they are due to have their garbage picked up every week.

Because of these security concerns, it's very, very common these days for people to lock up their garbage. Something they didn't even think about previously. But obviously, when you're throwing data out and that data can be used by someone else, you want to make it so it's very, very hard for them to get their hands on that.

There will be a fence. There will be barbed wire. There will be a lock. It will be very, very difficult to gain access to that one would hope.

You also want to consider shredding your documents. This is, of course, only going to go so far. It is not uncommon for somebody who is very, very dedicated and wants to be sure they're seeing information to take the shredded documents and unshred them. It's a big jigsaw puzzle. They'll put back together all of the documents.

A number of our more modern shredders pulverize this into tiny little pieces of dust. So you may want to think about making sure that if your data is being shredded that it's being shredded as finely as possible so that someone can't reconstruct it. The US government certainly realizes this so they'll burn it.

They just have burn bags. They throw their very, very important thing in the bags and the bags go out and they get incinerated so that nobody could ever possibly rebuild and reconstruct what was thrown out with the garbage. So you may want to go look at your trash. What's inside of your garbage in your organization?

Go down there and poke around. Grab some garbage yourself. Open up a bag. Is there something inside of there that could hurt your organization or that somebody else could use to gain access to people or resources inside of your organization? Dumpster diving is a very, very easy way to do that.

**Tags:** certification, comptia, dumpster, rubbish, security, trash

**Category:** CompTIA Security+ SY0-401

### **Tailgating – CompTIA Security+ SY0-401: 3.3**

Once you're inside of a building, the security posture of an organization is dramatically decreased. In this video, you'll learn how the bad guys can get into your secure building without being noticed.

If your organization requires people to walk into the building and be badged in as they're walking through as a security concern then tailgating is something you need to be aware of. With tailgating, we're using someone else to get access to a building.

Obviously, if I'm not part of your organization I don't have a badge. It won't allow me access in the door. So I need to find somebody to open that door for me and to allow me in.

And the guys who are coming in with this tailgating methodology aren't doing it as an accident. They want to get into your building. And this is one of the easiest ways to get in and make sure that they can go undetected through your security.

In the book, No Tech Hacking, Johnny Long gave a very good explanation of how you can use tailgating to get into a building, and all you really have to do is plan just a little bit. What he did was get the same clothing that a third party telecommunications company would use to get into the building. He even created a special badge that looked just like a badge from a third party telecommunications company.

And then he came in and showed up and made sure he had a legitimate reason to be there. If you are a company that has telephones then obviously, a telecommunications company makes sense. You would be in the building checking the phones, checking the wiring, that type of thing. He also temporarily took up smoking. He's not a smoker, but he realized that people are always coming in and out of the smoking area of a building. So if he can sit out there when nobody else was there and show up, he looks legitimate.

He looks like he should be there. It looks like he just walked out of the building to take a smoke. And all you have to do is now wait for somebody to let you back into the building as they are now returning from their smoke break.

I personally prefer bringing doughnuts. Instead of smoking cigarettes, I'll show up with boxes of doughnuts.

My hands are full. Please let me in the door. I'll try to catch people just as they're coming in in the morning and who wouldn't let a guy walk in with a box full of doughnuts. Maybe that's just me.

Now what you're inside, of course, very little people can do to stop you. There are no more badges internally. And if there are, there are very few badges on a floor or part of a floor of a building. Most of the security with this is going to stop right at the border.

You can't badge people going from door to door inside of an organization or certainly, not from cube to cube inside of an organization. So at some point, you can have access to either a large or a certain size area of information. And that's going to be very, very valuable. If you're already dressed the part, you've already got access, now you can walk wherever you'd like.

To stop this type of tailgating activity, there needs to be some very specific security based around this. There needs to be very, very big penalties for somebody who's going to let somebody in the building without a badge that hasn't signed in or that is not escorted. You should be able to look at any one and see their badge and make sure they either have an internal badge for the company or they have a visitor badge for the company.

And if they don't, you should ask them. You should be enabled to do this. This happened in an organization I was with.

I had a visitor badge. I put it on my jacket and I left by jacket at my desk to get a cup of coffee. While I was getting coffee somebody asks, I don't see a badge. Who are you?

Now granted, I was in the security department. They were certainly keeping an eye out for that, but it should be the same no matter where you happen to be. In fact, it should also be that if you're walking through the door it should be one scan and one person.

Sometimes this is one where people have to scan, walk through a door, and everybody else has to wait while that next person scans and the next person walks in the door. And if it's a manual door this could take some time. That should be part of the security policy.

Sometimes you have these mantraps set up. These mantraps are designed so that you badge, then the door will swing and allow you into the building, and you must badge and everybody goes in one at a time or comes out one at a time to be able to get in and out of the building. And you don't have a choice in that particular case.

Those types of mantraps or air locks are designed to make sure there is no tailgating. It's very hard for two people to squeeze into that very narrow area. And if you did, it would be very, very obvious that two people were squeezed into that.

So you shouldn't be afraid to ask. If you see somebody in your organization that doesn't seem like they should be here or they don't have your visitor badge or your company badge on them, it's a great way to make sure that they didn't tailgate to get inside of your building.

**Tags:** certification, comptia, security, tailgating

**Category:** CompTIA Security+ SY0-401

### **Impersonation – CompTIA Security+ SY0-401: 3.3**

Impersonation is the foundation of social engineering. In this video, you'll learn how the bad guys use impersonation to circumvent your security technologies.

A bad guy who is very, very good at social engineering will also be very, very good at impersonation. He wants to pretend that he's somebody he's not, whether he's walking into an organization by tailgating in and pretending he is somebody with a telecommunications company, or whether he's on the phone to you pretending he's somebody with the help desk. They now have a way to pretend that they are somebody that you should trust. And that is really the key.

You're getting information from your dumpster diving. You're getting information from some of the phishing that's gone on or from a third party, and you're calling up and saying, hi, I'm from the help desk. Hi, I know who you are. Hi, I work in this building or you're giving some specific details that would provide some level of trust.

You want to also consider that the people impersonating may be getting these attacks and they may be talking to you as somebody who's higher in rank. Well, I'm your boss' boss. I'm in charge of the entire internal audit organization so you better help me with this information that I'm looking for. So don't be intimidated by these things if somebody calls. There's nothing wrong with verifying that. And if they are from internal audit, maybe they'd appreciate somebody would be checking in on who that might be and make sure that they are legitimate.

Sometimes if you're calling you can throw a bunch of technical details around. Well, we're having problems with your computer because we're having catastrophic feedback due to the depolarization of the differential magnetometer. So we need to resolve this issue so I

need your password. So if you're able to confuse somebody with a lot of technical jargon can sometimes get them to oh, I had no idea there was a depolarization of the thing. Here's my details for logging in so maybe you can fix that.

And of course, just be a buddy. If you want to impersonate somebody, talk about what happened yesterday. How about that problem that we had in the building? Hey, did you watch the game last night? Did you see our local team that did this particular thing? And it makes you sound like you're right there in the building with them even though you may be thousands of miles away, trying to hack into their organization.

The bad guys will try to fool you into giving up your personal details. So as a rule of thumb, never give out personal information. Don't give out your username. Don't give out your password. Don't give out telephone numbers or email addresses. It's just something you should keep in mind when somebody's talking to you over the phone.

Somebody from the help desk doesn't need your password. They have access to whatever they need to gain access to without having your specific login credentials. Also don't disclose any personal details about where you work, departments, what you do, name, last name.

Those types of things can be used later on when you can call the next person on your list and say, I was just talking with Mary in accounting. She was telling me that she was having problems as well. And by gaining more information, it can make somebody else trust them just a little bit more.

You should always verify. There should be third parties, an intranet page, a phone list. Oh really? Let me call you back at the help desk and we can take care of that. Let me call you from another phone.

So you can try to verify this based on these people that are calling. And if you can call them back and verify that it's an internal number, it's something you can verify, then you can trust the person you might be talking to. In most organizations, verification should be something that's encouraged.

It should be part of your security policy. It should be part of the normal things that you do. And if you have this as a standard set of operating practices in your organization, nobody's going to be mad that you were checked on to get security. It's something that's accepted. And if you're going to stop the impersonators, you need to set up this corporate culture of verifying before any private information is given out.

**Tags:** certification, comptia, impersonation, security

**Category:** CompTIA Security+ SY0-401

### **Hoaxes – CompTIA Security+ SY0-401: 3.3**

We spend a lot of time and resources dealing with electronic hoaxes. In this video, you'll learn about hoaxes and some resources that you can use to research suspicious email hoaxes.

Hoaxes are messages that are presented to us. They're emails that we see that look like they could absolutely be real. And maybe they are something that tells us that we're going to be getting money. Maybe they are a hoax that says there's a particular virus or worm or security concern we should be aware of. But it's not actually real.

And because of that, it can consume a lot of resources. Usually, it's hitting something very particular about an organization. If you're running Windows, here's something you need to be aware of because everybody's going to be attacked by this worm. When in reality, this worm doesn't exist. There's not a vulnerability associated with it.

And we get it through emails. And people will see the email and forward the email to the people they know who will forward the email to people they know. They'll print it out. They'll put it on boards in the organization.

They waste time. They waste resources. When we really could be doing something else to help really protect our systems and our environments.

And we'll see this come in as an email. These days they come in as a tweet, as a Facebook post. They're now— any way that we can see information coming in, we can see this pop up on these messages.

And these Facebook posts and tweets bring a personal level to this. With social engineering, we trust our friends. And if our friends are telling us about the hoax, well, then it must be really. It can't possibly be fake.

Some hoaxes will take money. It's not just a security concern. They're telling you to send information and money right now to solve a problem, to get money brought into the states, to help somebody who's stranded overseas. Maybe they're hitting you through Facebook so you think this hoax is something you should be giving up your credit card information and wiring money across the pond. And in reality, it's all a big hoax.

If this is something that is a big hoax that really hits people on a particular nerve, it could be a hoax about a virus that ends up spending a lot more time than an actual virus might be. So we have to combat this all the time and make sure people understand that when they get an email, they have to verify this information.

Hoaxes about viruses are not viruses. We shouldn't be too concerned about that. The security people have ways to double check if the virus is actually a legitimate concern and have ways in place to prevent that virus from hitting our organization.

In fact, if you see an email come through that starts with, this is not a junk letter, well, then it's probably a junk letter. It's probably not real at all, especially when it tells me that Bill Gates is sharing his fortune and all I have to do is forward this email to friends and I'm just going to get a check from Bill Gates. What could be easier? Let me forward the email around.

That's a very benign type of hoax, but it's one that causes people to read it, waste time on it, forward it to other people, and it gets forwarded to other people and it sits in an inbox and takes up disk space and we have to back it up and it's archived, et cetera, et cetera. So if we can get rid of the hoaxes then we can solve a lot of resource problems in our organization.

So keep in mind that if you get a message and it says that I have won money, don't believe it. Believe nobody. It's the internet. Hardly anything I'm getting in my inbox tends to be real and honest and true.

So I should naturally look at this with a little bit of skepticism. I should not immediately take it at face value. There are a lot of sites on the internet where you can cross reference this. So go to [snopes.com](http://snopes.com), go to [hoaxbusters.org](http://hoaxbusters.org), go to [hoax-slayer.com](http://hoax-slayer.com) and cross reference this.

If it's worded and it looks like it could be legitimate then, of course, we should run it by some third parties and see if this is really the case. If you have spam filters, they may be able to look at this and immediately know, no, this is spam. I should throw it out. You'll never see it show up in your inbox. There's a lot of ways that you can filter this out so that it never appears in front of people's eyeballs so that then they would want to do something with it.

And if it sounds too good to be true, that I simply get an email from the Swiss lottery saying that I've won \$750,000 pounds, well, I probably have not. I haven't registered in any lotteries. I don't win the \$750,000 pounds. And don't you think they'd pick up the phone and call me? Why would they send me an email, especially knowing that most of our email is going to go right into our spam folder anyway.

**Tags:** [certification](#), [comptia](#), [email](#), [hoax](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Whaling – CompTIA Security+ SY0-401: 3.3**

These days, the bad guys know that the management of the company is the one with the money and the power. In this video, you'll learn about techniques that the bad guys use for hunting the big whale.

Why would the bad guys spend time trying to get information out of me when there are some much bigger fish in the sea? And that's where we get the term whaling. It's a lot like fishing, except now we're going after some really, really big fish.

So if you have executives in your organization, the executives undoubtedly have access to private information. They have access to internal information in the organization. If they are a public company, they probably have access to information that the public is not going to see until a big release of news. So this may be some really good things that the back guys are trying to get access to.

Usually, you're trying to get access to executives so you need some very, very specific information to gain the trust of these folks. They are up at a high level in the organization because they are very good business people. And they understand that people are going to be going after this very delicate or very sensitive information, confidential data, within the organization.

So the bad guys are going to send very focused emails. They're going to pretend to be somebody that your executives trust. They're going to have names and information that would cause the executives to feel at ease with the emails or the voicemails or the messages that they're going to be receiving.

A good example of this is what happened in 2007. In October 2007, a lot of financial companies were getting a lot of very specific information being targeted right at them. And not just one financial organization, but the executives of many different financial organizations.

So once you had the FBI and other organizations start looking into this, they realized—and this was announced in November 2007—that [salesforce.com](http://salesforce.com), which is a customer

relationship management front end on the internet, they store a lot of customer data. They were the victim of a phishing scam. And they had the internal people within Salesforce provide data, confidential data, to the bad guys. And that confidential data had the emails, phone numbers, and other personal information for executives, among other things.

So that guy said, forget all the other people. Let's just go after the executives. . And they started targeting them with these very specific names, very specific emails. It was all up to date because it came right out of the salesforce.com database.

Because of this very targeted type of attack and the fact that it's using some very well-known communications means it's very, very hard to prevent this type of whaling attack. It goes right through your firewall, right to your IPS, right through your email filter because to the human eye and the technical things we have in place, it looks like an absolutely completely legitimate message. It becomes very, very difficult to stop it through traditional means.

So that means we have to train our executives. We have to make sure they understand what the bad guys are doing to gain access to this data, and make sure that they know what they should be looking out for. Executives, generally, are very mobile. They are in your building. They are out of your building. They are carrying **iPads** around and **iPhones** and other mobile devices so they might have some very unique security concerns that other people in your organization just don't have.

So make sure that you are modifying and keeping up to date with the latest technologies because you're going to need to apply this security to everything that the executive is doing wherever they might go. And of course, you might want to test this.

You might want to have your CIO phish the CEO and see if the **CEO** is going to bite. So do some internal testing, some internal auditing. Get the CIOs and the other executives accustomed to somebody coming to them that they don't necessarily know very well and asking for very, very important and confidential information. The more work you do up front with your executives, the less opportunity the bad guys are going to be able to score on this phishing and this whaling expedition.

**Tags:** certification, comptia, phishing, security, whaling

**Category:** CompTIA Security+ SY0-401

### The Effectiveness of Social Engineering – CompTIA Security+ SY0-401: 3.3

Is social engineering really a threat? In this video, you'll learn how a talented social engineer was able to steal a valuable Twitter handle.

Social engineering is sometimes thought of as an attack type that's not very critical, but what we're finding is that the bad guys have gotten very good at obtaining information from us and they're now going after some very high target types of information, data, and assets. The bad guys have figured out that if they want some information about you, they may just need to go to multiple organizations to gather that.

They can go to the places you shop. They can call in to the places where you have accounts. And they can start gathering details or finding access in to get information about you or about your financial situation.

Sometimes they are using this social form of communication to be able to manipulate people on the other end of an email. They might be calling in and being very aggressive on the phone or they might be sending a message about a funeral notice for someone you know. And, of course, you want to click on that. You want to read the attachment, which is exactly the social engineering attack that the bad guys are going after.

One of the more recent and interesting social engineering attacks is one that took place against Naoki Hiroshima. This is how I lost my \$50,000 Twitter username. You can Google that phrase or go to this URL to read all about it.

This is an example of how the bad guy was able to use multiple organizations against this particular person. And it was really masterful social engineering that these people were able to pull off. The first thing the bad guy did was call Paypal and he used social engineering against Paypal to obtain the last four digits of the credit card on file.

Now obviously, the last four digits of a credit card are not going to be used for charging anything. You can't really do anything financial with the last four digits. Or can you?

At that point, the bad guy called **GoDaddy**. Obviously, the bad guy had done a lot of research on Mr. Hiroshima and knew that all of his domains are being hosted at GoDaddy. So he told GoDaddy he lost his card, but he could tell them what the last four digits of his card were.

And GoDaddy said, well we need the first couple of digits as well. And unfortunately, instead of simply identifying right then the first two digits of the credit card, GoDaddy allowed the person calling in to guess and to keep trying until they got the right two digits. This is social engineering done extremely well. And unfortunately, this allowed the bad guy to gain access to all of the domains that were hosted at GoDaddy.

So now Mr. Hiroshima has no access to the domain names that are registered at GoDaddy. The bad guy now has complete control over all of those things. And has changed all those security parameters so that Mr. Hiroshima cannot go in and gain access to those.

He then contacted Mr. Hiroshima and says, I'll tell you what. I will give you access back to all of these domain names. You just have to give me your Twitter name, which in this case was an @N, that single letter N. So that was relatively valuable for the bad guy to have. And in this case, there was an agreement and the exchange and the swap was made.

Mr. Hiroshima then went to Twitter and asked them for help because obviously there was extortion involved and fraud involved at obtaining this @N Twitter username. And

ultimately, it took about a month. But finally, Twitter said, yes, you are the rightful owner of this username and we're going to restore access back to you.

So ultimately, he was able to get both of his domain names back. He was able to gain access to his Twitter handle, and finally had everything back the way it was. It's an amazing story of social engineering.

If you'd like to read about it, you can Google the phrase how I lost my \$50,000 Twitter username. Or go to that URL, and you'll see just what the bad guys will go through to get the information that they really want.

**Tags:** certification, comptia, security, social engineering

**Category:** CompTIA Security+ SY0-401

### **Rogue Access Points and Evil Twins – CompTIA Security+ SY0-401: 3.4**

One rogue access point can create a significant security issue. In this video, you'll learn about rogue access points, evil twins, and how to protect yourself from these security concerns.

A rogue access point is, quite simply, an access point that's been added to your network without your knowledge, you no idea it's there. This is obviously something that can create a very significant backdoor. If you don't know an access point's there, then you certainly aren't managing it, you don't know if any type of security has been configured on it, and you have no idea who might be connecting to your network through this wireless connection. So there's some really, really huge security concerns associated with the rogue access point. The problem is that it's so easy to plug-in an access point into a traditional network. If you're not doing any type of network access control protocols on your network, then it's very easy, not just for workstations, but for things like additional access points to be plugged in to any network connections. Somebody can walk into their cube, plug-in this access point, and now they're on the network with this access point.

One of the things that's also a bit of a challenge is now the latest operating systems also allow you to click a few buttons and perform network sharing using existing wireless connections. You can then plug into a wired connection and have your computer become the access point. The wireless card in your computer is now its own broadcasting access device. This is very great when you're on the road and you like to share that connection, not so great when you are in your corporate or organization's environment and you want to be sure that nobody can connect to the network who should not be on the network.

Obviously, to be able to combat this you either have to have some type of network access control in place or you may have to occasionally grab a wireless access point device—something that can survey the area—start walking around. See if you can find access points on your network that you have no idea are really there. There are lot of great tools to do this. There are commercial tools and, of course, tools that you can get for free from the internet that would allow you to see what's happening in your wireless network, as well.

If you have the flexibility of enabling network access control, which are these 802.1X type protocols, then you're requiring that people authenticate to the network every time they plug-in a device, whether they're plugging in and connecting via a wireless network or through a wired network. Now this won't necessarily prevent people from plugging in an access point, but what it will do is require that people who are connecting to that access point authenticate through the methods that you have in place. So even if they were to some way connected physically to the network, they would still be prevented from doing anything in your environment.

When the bad guys are putting together a rogue access point it's for much more nefarious means. They really want access to your network. Or they want access to what people are putting through the network. And that's where a wireless evil twin comes into play. Very simple to do this. You grab an access point, you purchase one. In the United States you can get one for well under \$100 these days. You have this access point, you plug it into the network, and you configure it exactly the same way as the existing network. This is why open access points that have no password associated with them can be such a security concern, because it's very, very easy to duplicate an open wireless system. You simply put the same configurations in the evil twin— the same **SSID** information, the same security settings. If you do have access to the security settings, simply duplicate the security settings on the evil twin.

And once you're ready to implement it, you put it into the network or position it in a place on the network so that it is the primary excess point. You generally do this by making sure that it is the one with the strongest signal for the end users to see and the machines will automatically see that stronger signal and decide to choose that access point. It just makes sense. So normally you're trying to get much more power out of your access point than the existing access point that's on the network, or you make sure the evil twin is closer to the people that you would like to gather information from. And once you're on the network, you're able to see everything. It's very, very easy to do this in a place where there are open Wi-Fi hotspots.

The challenge at this point is now you're connected to the evil twin, all of the traffic going between you and the regular network is now going to flow through the evil twin. Which means anybody who has control over the evil twin can see everything going over that link. This obviously creates enormous security concerns and as we go through these videos we'll talk about things that you can do on wireless networks to look for these things like the wireless evil twins and what you can do to protect your data in case you happen to be connected to one of these evil twin networks.

And one of the most useful functional ways to protect this is to encrypt your data. Even if it is over a wireless network that already has **WPA encryption**, you want to even add additional encryption by perhaps creating a **VPN connection**— a tunnel that is an encrypted connection between your machine and another device. Maybe you're using **HTTPS** to your web servers, you're logging on through an encrypted channel that goes between your workstation and the web server on the other side. Even if somebody had an evil twin on the network, they were monitoring the traffic flowing across that wireless network to the wired network, they would not have any idea what was inside of that traffic because you are protected. You're encrypting all of that traffic and even if they capture it, they can't do very much with it.

**Tags:** [certification](#), [comptia](#), [evil twin](#), [rogue access point](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Wireless Interference – CompTIA Security+ SY0-401: 3.4**

Our wireless networks are only as good as the signal it provides. In this video, you'll learn how the bad guys are disrupting our wireless infrastructure to help them gain access to our network.

One of the challenges with managing connectivity through a wireless network is that these are wireless signals and they can be corrupted, they can be interfered with. Even though it is a digital signal, you still have things that can create problems for you when you're trying to install wireless connections, when you're trying to maintain wireless connectivity, and if somebody's trying to disrupt what's happening with your network communication making sure that they can interfere with the wireless signals is a great way to do that. This is a bit of a problem because there's so many devices in our networks, both at work, at home, anywhere we happen to be, that can create this interference— a Bluetooth headset, a microwave oven, a cordless phone— they all can create signals that can create problems for us to receive and transmit traffic through our wireless networks.

If you're in the United States, it is part of our federal laws that it is illegal to disrupt or interfere with these wireless signals. And a lot of people would like to do this, they'd like to have in their environment to prevent anyone from using a wireless network, to install a jammer inside of their environment that would prevent that. However, those types of things are illegal because interference with these wireless signals creates problems not just for your wireless network but for other wireless devices as well. And since a lot of this revolves around public safety, it's something that is absolutely illegal to have. There are illegal jammers that are out there, they are a violation of federal law in the United States and they're probably a violation of a number of laws elsewhere around the world. We have situations where your signal degrades, however, and it may be something that is external to what you're doing.

If you are degrading the wireless connection from an attacker's perspective, you can degrade it in a way that people are not able to operate. And if you're trying to create a denial of service situation, interfering with this wireless signal is a great way to do it. It may be absolutely illegal to do this, but the bad guys generally don't care so much about the federal laws. Sometimes you'll see this being used in the case of where there is an evil twin. In our previous video, we talked about wireless evil twins and how they're used. If I can knock out the wireless signal to the primary access point to the real access point, then suddenly my evil twin now becomes the standard connection to the network. It's yet another way to create a denial of service on one access point and have my evil twin become the actual access point that's now used for everybody.

If you have a way to plug-in some additional devices, like a spectrum analyzer, you can see the exact signals that are on your wireless network and the type of traffic that's there. So if you're wondering if somebody might be interfering, or if there's a third party, or some other device that's interfering with your wireless network, spectrum analysis can be a great way to narrow down exactly where that's coming from. Obviously finding this signal that's disrupting things is your primary goal. And if you have a spectrum analyzer, it's a great way to narrow that down. You can do some of this analysis if you have some simple software that examines traffic.

Maybe you're just finding a rogue access point that's somebody has set up. But if this is not a normal 802.11 communication, it may be just a blast of signal and has nothing to do with 802.11 communication, you're going to need a spectrum analyzer. And as you can tell, these spectrum analysis software and hardware combinations are not the easiest thing to understand. It may take a little bit of work and a little bit of training to see exactly what all of these things mean as these different colors and different lines are going across the screen.

You can also take your existing access point and sometimes there's an option to boost the signal. This is something that's not available on all access points, but generally the access points that are in a large environments, in corporate environments, those types of access points tend to have this flexibility so that you can adjust the exact signal you would like. And if there is another device out there creating interference, sometimes boosting the signal can allow other people to hear what that access point is saying. You also might want to try different frequencies. If the bad guys are focusing on taking your access points down by creating interference, they may have only selected a certain narrow band of frequencies just to take your access point down and perhaps not someone else's.

So one of the things you can do if you have spectrum analysis, you can see exactly which frequencies are in use. And if you're down at the lower channels, you may want to set your access point to operate at higher channels away from the offending interference. This maybe a cat and mouse thing. You may be moving back and forth. And in the meantime perhaps you can hunt down where the offending signal is coming from. And if somebody is maliciously trying to create interference so that your network isn't going to operate, that may be your primary goal to be able to find that, get it off the network, and allow your network traffic to communicate normally.

**Tags:** [certification](#), [comptia](#), [interference](#), [security](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Wardriving and Warchalking – CompTIA Security+ SY0-401: 3.4**

We've rapidly moved from chalk-based wireless network identification to completely automated wireless maps. In this video, you'll learn about the history of **warchalking** and how today's wireless crowdsourced mapping is accomplished.

If you're wondering how the bad guys find your access point to begin with, well, it's simple. They're looking for them. But in some cases, they're not just targeting you. They want to find access points wherever they might be. So there is something called wardriving that's very common. You'll find people hopping in their car. Now your Wi-Fi systems have very nice antennas you can connect to them. Many times there's a **GPS** functionality that you could plug into this as well. You hop in your car and start driving around. In fact, you don't even pay attention to what's on the computer. Just drive around for while. The **GPS** is going to be monitoring where you go, and you'll find that you collect a huge amount of information over a very short period of time.

If you're using different software to find these access points— here's one on my Mac called **Kismac**— and you can see just sitting in one place in my house, not even driving around, you can see all of the different access points around you— the ones not only in your house, but elsewhere. Now imagine driving around your entire neighborhood, around your entire city. You may be surprised at exactly what you're able to see.

And the big challenge for us as security professionals is the ability to do this. It's very, very, very easy. All of these tools are absolutely free. There's something called Kismet. I'm using the Mac version of that called **Kismac**. There is the wireless geographic logging engine you'll find at wgle.net net which allows you to take all of this data that you've created that you found by driving around and put it into a geographical database. It doesn't have to be by car. You could also do this by bicycle. You could do this by connecting a wireless access point to a radio-controlled plane. This actually happens. And you'll have people who go up in an actual plane and fly over an entire city to be able to see what's going on. This is not unheard of. So your access points now become a data point in the big database that the bad guys are using to figure out where can they gain access to a wireless signal.

The results of all of this geographical data and GPS data, wireless information, all being combined together can create some very dramatic views of the world. You can start to see where people are finding access points, where there may be closed access points, open access points. And you can see exactly where you drove and where those access points were and how many you found. There is an amazing wealth of information out on the internet, some that's being used for good, so that we know where these access points are and where you might be in relation to these access points, and some that might be used for bad. If you're looking for open access points and ways to break into people's networks, this is a very, very simple way to look at a map, go to a place, and see exactly what's going on without wireless network.

Back before we had GPS connectivity and a way to drive around with a wireless device, there was a laptop. It was able to gather this information over long distances. We did something called warchalking. Or at least it was something that was mentioned. These days, it's more of a historical footnote to how this whole wireless network generation came about. It used to be that we didn't have this technology. There weren't signs on the door that said there was free Wi-Fi. The people that had an access point that were plugging into the internet were very unusual to find.

And so there was this concept created called **warchalking**. If you were fortunate enough to find somebody who had a wireless network, this gentleman called Matt Jones created this set of symbols. So you could look on the sidewalk. Someone would have written in chalk or on the wall. Someone would have written in chalk some of these symbols to note whether there was an open node, a closed node, or one that was wireless encrypted. It even was the **WEP node**. There was no **WPA** back in those days. That way, if you happened to find one that was available, you could tell the world. You could say, oh, I found an open access point. Let me draw a big symbol on the sidewalk. Let me draw a big symbol on the wall, and let everyone else know.

Well, the situation, of course, since those days, has changed dramatically. These days it's hard not to find a wireless access point that you can use. Almost everybody is putting them on their window, saying, oh, come in our place. Buy some food, buy something to drink, get some coffee. You can hop on our wireless access point. They're using it as a sales tool, a marketing tool, to get you in the door. So although this warchalking isn't used anymore, it's really just migrated into another way that we can use to find out where can I go to get a wireless signal.

**Tags:** [certification](#), [comptia](#), [map](#), [security](#), [warchalking](#), [wardriving](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Bluejacking and Bluesnarfing – CompTIA Security+ SY0-401: 3.4**

We rely on Bluetooth networks to connect our mobile device to headphones, speakers, and other devices, but is a Bluetooth network really safe? In this video, you'll learn about the security concerns surrounding **bluejacking and bluesnarfing**.

The Bluetooth technology that's now in our mobile devices is a great technology. It's there and we can communicate with our computer, we can communicate with other mobile devices all with this Bluetooth technology. It even allows us to have these headsets that we have sticking out of our ears so that we can have these conversations without having wires anywhere.

The problem, of course, is that any time you have a wireless network there's an opportunity for people to abuse that wireless network. And so there has been something called **bluejacking** that does something like that. This is the ability for you to send an unsolicited message to a Bluetooth-enabled device. This is spam for Bluetooth.

It's not an email message. It's a little bit different than that, but it's a way that without anybody else's authorization you can send them a message. They can see things pop up on their screen. It becomes a bit of an annoyance.

And it does this because we have this network here. **Bluetooth** is a great network. It's one that is immediately available to us. We simply turn it on and we can take advantage of it.

But it does have a limited range. So to be able to do bluejacking, you at least have to be in the range of Bluetooth, which generally is about 10 meters, sometimes a little bit more, sometimes less depending on the interference, but it's one where somebody is nearby. Now if you're in a city, you're in an airport, there's a lot of people nearby with Bluetooth. But obviously, if you're sitting at home it's hard for somebody to get that close to you to be able to send you these messages.

The way that bluejacking works is it takes advantage of something that is a convenience for us on our mobile devices. And the convenience is being able to communicate back and forth and send things, like contact information between the devices automatically. So if I meet someone for the first time, I may want them to have my contact info and I simply tell my phone, please send my contact card over to that Bluetooth device and it sends it. The other device is waiting for it. It simply receives that Bluetooth message.

Now you can then, if you use that same methodology, get your own messages sent to a Bluetooth device because there's no authentication. There's no extra authorization that has to be done. You can simply send these messages to a Bluetooth device.

So the way that the bad guys do it is you create an address book object, you create a contact in your contact list, and in the name you put the message that you'd like to appear on that Bluetooth-enabled phone. So instead of putting James Messer so that the message popping up on the other person's phone says James Messer has sent you some contact information. Would you like to accept it? It says, you are bluejacked or I'm taking over your phone or something that might scare somebody when the message pops up on their screen.

So they see the message pop up that says, you are bluejacked. Add to contacts? Now obviously, all I'm doing is sending a contact message. There's nothing in there that is harmful. I'm not stealing somebody else's information.

It's more of an annoyance. It's like spam. But they have to read it. They have to see it on their screen.

And so there have been cases where even advertisers have created methods where they'll send these Bluetooth messages to people walking by a store or walking by a display. It is annoying. It is spam like. It's something we didn't ask for and therefore, it's something you just have to keep in mind.

There's third party software that can do this. Something like **Bloover** is a good example of that. You can download **Bluesniff**. These are Linux-based front ends that allow you to send these Bluetooth messages to other Bluetooth devices and annoy the person that's on the other side.

This is a little bit different than something malicious because **with bluejacking** we're simply being annoyed by messages. Unfortunately, there have been cases where Bluetooth has created more of a problem for us because people were able to steal our information. We call that method of stealing our info **bluesnarfing**. Somebody is using Bluetooth to **snarf** our data, to take our data right off of our phone.

This is when a Bluetooth-enabled device is able to use a vulnerability in the Bluetooth networking to be able to get onto a mobile device and steal contact information, email messages, pictures, anything you might have in a file on that phone. Different phones work different ways with different file types, but if you know the phone type that's there or you know what to look for, you can essentially download things directly from somebody's Bluetooth-enabled device without them knowing.

Now that, obviously, is a little bit of a challenge if you have a Bluetooth device. But fortunately, this is a very old vulnerability. It was found by **Marcel Holtmann** in September 2003. Adam Laurie also saw it in November 2003.

And both of them worked with the different phone carriers, the different phone manufacturers, and of course, the Bluetooth alliance to be able to patch this particular weakness in the Bluetooth software and the Bluetooth networking functionality. Those protocols themselves were a problem. So by patching it, the problem went away.

Obviously, this is a problem that occurred a long time ago. 2003. All relative to our computing devices. We go through our mobile devices every two or three years. We update the software on our mobile devices so it's very hard to find a mobile device still running one of these very, very old versions of Bluetooth.

But this speaks to a bigger problem, of course. We have so many mobile devices. We have so many devices that use different mobile technologies and different networking technologies that we as security professionals have to stay up to date. And whenever something new comes out, we always have to look at it with a skeptical eye to determine if this is something safe to do because you don't want to run into one of these situations where you have bluesnarfing where somebody's able to take advantage of a vulnerability in the basic functionality of these networking protocols to be able to steal information from us. And so it's very important that we just keep an eye on it and be able to understand what has happened in the past that we can then prevent those things from occurring in the future.

**Tags:** [bluejack](#), [bluesnarf](#), [bluetooth](#), [certification](#), [comptia](#), [security](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

### Wireless IV Attacks – CompTIA Security+ SY0-401: 3.4

Many encryption methods use initialization vectors to provide additional randomization to the data. In this video, you'll learn how a poorly implemented initialization vector created an enormous security concern for our wireless networks.

Initialization vectors are a very common mechanism used when dealing with encryption technologies. Whenever you're sending information back and forth between devices, whether it's over a wired network or especially over a wireless network, you want to make it as hard as possible for a bad guy who's sneaking in and looking at that information to be able to decipher that data that's going back and forth. And if you're using the same key to encrypt this information and send it across this network every time you're sending data, you're making it very easy for the bad guy to go through his algorithms rhythms to decrypt that information. And that's where initialization vectors come into play. If you can change the key every time and yet have the key exactly the same, you make it very, very difficult for people to be able to decrypt this.

So that's what the IV, the Initialization Vector, does. It is added to the key to essentially create a scrambled up or different key every time. And it's done in a way that the station on the other end is able to also know the initialization vector and essentially undo what was done with that. Makes it very, very easy to send information over a network that's encrypted, but have it different every time that data is sent. Ideally, this now becomes very, very protected data. But as we're about to find out, it doesn't always work exactly the way we might have planned.

And if you're aware with 802.11 with WEP, encryption even though it was using initialization vectors, we created a bit of a problem. It was not implemented in a way that was protecting our data in the best possible form. If you're trying to make sure that nobody's able to get into this connection and see this encrypted data, then you want to be sure you're using a mechanism that's very, very strong. Unfortunately, with 802.11, we found that the cryptographic algorithms that were used as they were associated with these initialization vectors weren't strong enough.

There were a number of technological challenges that really created this issue for us with 802.11 WEP. One was that the federal government of the United States said, you can't do heavy encryption on these wireless networks. We would not be able to look into that if you did. And so it limited the key sizes you would be able to use to encrypt this data. The initial key sizes that you had were only 64 bits in size. Later on, we got that increased to 128 bits, but at that point, it became a little bit more of a problem. We'll talk more about some of those issues in just a bit.

So what we ended up having was really a 40-bit key. And because the initialization vector is also part of the key, we had 24 bits that were set aside for the initialization vector. So that totaled 64 bits. Let's look to see exactly how this process occurred. What you would have whenever you're encrypting data is you have some plain text. You have what you would like to send to the other side. And so we would take that data, and we also created a cyclical redundancy check for that data— a way that we could check on the other side that the data was not changed between point A and point B. So those two things together are what we would like to encrypt and send to the other station.

Now we also have this key that we're using to encrypt it with. We have our WEP key here, and we have our initialization vector. You can see these boxes are not to scale, so don't think that the **WEP key** is necessarily smaller than the IV. But those two things are put together. And a mechanism, a cypher, is used on those to encrypt that data called **RC4**. **RC4** is also something you can use also to decrypt the data very, very easily. And we'll

see why that's important in a moment. But those two things combined together. The **RC4** created a key stream. And the key stream and the combination of the plain text and **CRC** put through x or mechanism the finally creates the cypher text.

And on the other side, when we're ready to send it, we take the initialization vector, we connected to the cypher text, and we send it across the network. This is the mechanism that's used also on the other side. It receives the IV in the cypher text and knows what the IV is so they can then perform, or reverse, the x or. It gets a plain text in a key stream. It reverses the RC4. It knows what the initialization vector is, therefore it knows what the WEP key is, and therefore it's able now to decrypt what's going on.

This is also a challenge, of course, because we're sending the initialization vector in the clear. And there are also some challenges now making sure that this process of encrypting the data really is strong as it really should be. If you've ever worked with WEP or you know how to implement this wireless encryption on a wireless network, one of your first challenges is that everybody has the same key. There was no requirement in WEP that people's keys change over time or that people could have different keys. So everybody tends to have the same key. That means if somebody leaves the organization, they're taking the key with them. So maybe you have to now change the key. That now also means that you change the configuration in every single person's wireless station. And sometimes it's just not practical to be able to do that.

Another problem we found— this is more in the details of the cryptography— is that the initialization vector is only **24 bits long**. And in the big scheme of things, that's a relatively small number. That means that you only have just over 16,000,000 different possible iterations for an initialization vector, which means it's very, very common once a lot of data goes over the network, to see the same IV crop up again. And if you can get two data streams using the same initialization vector— because I know what that is, it's in the clear— then I can start comparing those two different pieces of encrypted data to determine the key that was used under the surface. This is probably something that should have been avoided to begin with, but because your key links were so small there wasn't much of a choice there.

Another piece that was a challenge is that there were certain initialization vectors that would not properly encrypt the data. It wasn't really giving you a very strong encryption. So certain IVs were creating what we call this weak type of initialization vector. Later on, there were certain devices, certain access points, and certain wireless cards that would not use those weak IVs. But unfortunately, there were devices that absolutely was. And if you could see some of those weak IVs coming across the network, you could then be able to discover what the key is, because things weren't being encrypted very well.

The bad guys love this. They thought, if I could then create a lot of initialization vectors and really go through a lot of them, I can create a lot of duplicates of those. I'll churn through 16.7 million of those a couple of times, and I'll start to see differences in the encrypted data as it's coming through. So they created software that would essentially just put a ton of packets on the network so that they could churn through all of those IVs and start to get duplicates and start to examine the duplicates. This made it very, very easy to find the key. And if you see some of the modern types of software used to do this, the WEP Crack software that you can download for free on the internet, they're able in some cases to identify the WEP key in just a few minutes. And that's why whenever we talk about securing your wireless network, one of the first things we always say is, don't use WEP. And that's why some of your new access points don't even give you an option for WEP encryption, because they know that there are so many attacks out there that can very, very easily take advantage of these problems within initialization vectors in the 802.11 WEP protocol.

**Tags:** [certification](#), [comptia](#), [initialization vector](#), [iv](#), [security](#), [wireless](#)

**Category:** CompTIA Security+ SY0-401

### **Wireless Packet Analysis – CompTIA Security+ SY0-401: 3.4**

There's a wealth of data hidden in the packets that traverse our wireless networks. In this video, you'll learn how easy it is to perform wireless packet analysis and what you can do to protect yourself on a wireless network.

One of the challenges you have on any network, whether it is a wired network or your wireless network, is that if somebody can get a packet capture from what you're doing then you'll be able to see a lot of what's going on. Everything going back and forth is in those packets. And that's why you'll find that a lot of security professionals go even overboard to make sure nobody's able to see what's inside of that data that's going back and forth. Unfortunately, a lot of what we do day to day is absolutely in the clear.

There's really not as much encryption going on over the network as you might think or you might hope. So it's very, very simple now when you connect to a network, especially a wireless network, to see an amazing amount of information. If you're in a wired network, it becomes a little bit more difficult to capture. You have to be in a physical place, you have to tap into the network in some way, you have to see what's going on. So it becomes a little more difficult, you have to have exactly the right location on a wired network. On a wireless network, however, wide open. You can do a lot on a wireless network to see what's happening.

This module and the CompTIA exam requirements is called wireless network sniffing. But just so you're aware, the term sniffer is a registered trademark. So it's a term that we have we have really co-opted in the security— in the network analysis realm. A more generic term would be network analysis, and that's why you'll see whenever I'm working with different devices I'm not using a sniffer, I'm using a network analyzer— just so you're aware of exactly the type of technology in play. Capturing information over a wireless network is painfully easy. It's so simple to see every bit of traffic going on over that wireless network. Especially if you're sitting very, very close to the access point on that wireless network.

Your device— your wireless card, your wireless adapter, your laptop computer— can hear everything going on in and in normal operation it only acts on the information that's sent to your machine but because wireless is such a broadcast mechanism, every device on the wireless network can hear everything that's going on. This makes it very, very easy to set your card up in such a way that allows it to see everything going back and forth over the network. One of the challenges you have, if you are trying to analyze network traffic, is you have to make sure that your network card does not send information at the same time. On these wireless devices, if I'm broadcasting I'm essentially overloading my local receiver. So when you start up a number of different manufacturer's software to be able to capture they're using a special driver that turns off the transmission feature so it's able to capture and hear as much as possible.

Sometimes your network drivers will not capture wireless information. They simply are not configured, or have the right chipsets, to be able to do that. So as you're looking through software that allows you to capture from a wireless network, that software will tell you exactly the type of wireless card or exactly the type of wireless chipset that's required. In fact the manufacture of the software may be providing you with their own type of network driver for that card because they're able to turn on that feature where your normal driver does not have access to do that.

Sometimes you get a driver or a combination of software that kind of puts you in the middle. You can't see the wireless communication, the wireless protocol going on, but you can see the ethernet data that comes out of the wireless traffic. So that might be

enough for you to be able to see what's going on but if you're trying to figure out channel information, encryption information, and other things going over the wireless network you need to make sure that you can capture the entire wireless packet. Otherwise you're just going to see the data once it's come off of the wireless network.

And of course you can try this yourself. You can go out to [wireshark.org](https://www.wireshark.org), that is probably the world's most popular network analysis software at this point, and download that. Load it yourself. Try it with your access point. Try it with your wireless network cards. There's also some documentation on the wireshark website that can get you started and tell you how you can optimally configure your wireshark configuration and your hardware to be able to gather as much information from the wireless network. It is amazing how much detail you can really pull out of a wireless network.

This is wireshark. Even see the different interfaces on my computer. There's one that even shows it's the wireless adapter on my computer. I'm just going to click that. And we're going to start seeing the wireless information go back and forth over my wireless network. I'm not really sending a lot of data over that connection right now but look how simple that was. I load up my software, I say go start capturing traffic, and it starts pulling in that traffic and showing all the information going over that wireless network. If you're sitting in a coffee shop, if you're sitting out at work, if you're sitting at a conference, especially in places where they are open access points, now you're able to see a lot of in the clear traffic going back and forth over this wireless network. And that becomes an enormous concern from a security perspective. So it becomes important also to know how you can prevent people from seeing this traffic going on over these open access points.

The first thing you can do is make sure that the data on your wireless network is encrypted. Even if you have someone with a packet analyzer that's capturing all the traffic flowing through the air, the only thing they'll be able to make out is a bunch of encrypted data within those packets. And if you're doing WPA2 or WPA, that data is pretty well encrypted, it's going to be very, very hard for them to determine the key and be able to get into that information. We talked in a previous module about how WEP is a very, very bad way to encrypt data. In fact your access point may not even have the option these days to allow WEP, it's just something that if you see a legacy access point that's using WEP you should avoid trusting that as a way to protect your information.

You also can use encryption. Make sure that you go to a website and you login on their **HTTPS page**. Whether this is Google mail or Yahoo mail or whatever website. Any time you're transferring information, you're adding a user-name and password, you're looking at sensitive data, you want to be sure you're on an encrypted web server. And of course one of your options is to create an entire encrypted tunnel through that wireless network to an end point somewhere else on the internet and send all of your traffic over that encrypted tunnel. That ensures that whether you're going to an encrypted web server or web server where information would normally be sent in the clear, you're encrypting that up, sending it through the tunnel. You would have to be on the other side of the tunnel termination point to even see any of that information. And at that point you're probably through the wireless network and somewhere else down on the line in the internet.

Some people also take advantage of some virtual tunnel networks like Tor, which stands for the onion router. Or they may be using something like **Ultrasurf**, which is a very, very easy to use encrypted proxy where you can send information back and forth. In any of these environments there are advantages and disadvantages to doing any of those, so you want to be sure that if you're using a **VPN connection** or you're using an encrypted proxy that you trust what's really going on on the other end of that communication. Because ultimately your data will come off of that encrypted tunnel, and the some of that will certainly be in the clear.

**Tags:** certification, comptia, packet analysis, security, sniffer, wireless

### **Near Field Communication – CompTIA Security+ SY0-401: 3.4**

Our mobile devices are used for communication, entertainment, and now for near-field applications. In this video, you'll learn about near field communication and how this may change our perspective on mobile security.

A relatively new networking technology that's important to everybody who needs to protect their information is something called **near field communication, or NFC**. This builds on the legacy **RFID technologies**, whereas **RFID** was one way communication, **NFC** is a two way communication. And usually it's between a mobile phone or mobile device and some other third party device that you're communicating with. As the name implies, there doesn't even have to be a physical connection, you just need to get near the other device so that those two devices can then transfer information between them.

You see this being rolled out today with something from Google wallet and MasterCard, they're trying different ways of enabling these payment systems where you simply wave your phone at a payment system like this and it will transfer the proper amounts to that. You could also see this in use when you're starting up something like a Bluetooth connection and NFC could start or bootstrap that process to make the Bluetooth pairing work much easier than it's been in the past. We could also use this to gain access to a room through a lock. We all have our phones with us, instead of carrying around an access card or a smart card, we can simply wave our phone at a door lock and gain access into the room that way.

From a security perspective there are a number of concerns dealing with the NFC technologies. First off this is a wireless network communication. So it is possible to capture that communication and then be able to do something with the information that was gathered. Although it's a very close communication that's required to complete this NFC, when you're in the right place with the right type of antenna, it could be as far as 10 meters away and still be able to see or hear the information that was transferred during that transaction. Another security concern is relating to denial of service.

If this is something that can gain access into a room, we could then send a frequency to jam the **NFC communication**. And although you were right there next to the door and you had your phone and it was very close, you still would not be able to gain access into those resources. Just like any other network, you also have the concern of a man in the middle attack. So the man in the middle could be relaying information between what may look like a **legitimate NFC** end point and the actual **NFC device**. Or it could be a capture of information and then a replay of that information into that wireless **NFC network**.

And of course, if we lose our phone and no longer have access to it, somebody may be able to gain access to our lost phone and then use that to gain access to a door or to pay for items using that **NFC technology**. So although in NFC does provide us with some ease of use when we're performing financial transactions or gaining access to resources, we still have to be aware of some of these security concern surrounding the technology.

**Tags:** certification, comptia, mobile, near field, security, wireless

**Category:** CompTIA Security+ SY0-401

### **Wireless Replay and WEP Attacks – CompTIA Security+ SY0-401: 3.4**

Flaws in WEP encryption were exploited using a series of replays and very specific cryptographic attacks. In this video, you'll learn the process that the bad guys used to break WEP encryption.

When you're on a wireless network, you still need to be aware of the possibility of a replay attack. This is very similar to the replay attacks you would get on a wired network. In fact, on a wireless network it may be even a little bit easier to gather the information that someone might need to then perform that replay attack. This is a big problem for people that are trying to protect their wireless networks in the enterprise because you're sending that signal out everywhere. It's very difficult to localize it, and therefore that opens up some security concerns for you as a security professional.

It's obviously easy to capture this data, especially on something like a hot spot where all of the information is naturally in the clear. You're not doing any type of WEP or WPA encryption, you have to require the end user to provide their own VPN or other encryption mechanism to be able to protect their data. And of course, not everybody on the wireless access point in these public areas is going to have that type of security in place. It's very easy for the bad guy to tune in the SSID of the wireless access point and listen to all of the information going by. That's a perfect place to gather information that you can then use for a replay attack.

The cryptographic problems of the WEP encryption protocol were really something that we were able to take advantage of because of a replay issue. The WEP encryption allows you to replay information using exactly the same key. And so it was very easy for someone to collect information and then send that information back out again.

To perform a crack of WEP encryption you needed to gather a lot of initialization vector packets. These are packets that normally are sent back and forth when a system is connecting to the wireless network, but when you're trying to gather thousands of these you need to create your own method to gather them. And being able to send ARP requests across this wireless network allowed us to build a large group of initialization vector packets and store all of that data. You can sit and either wait for people to do it themselves or take advantage of this replay of the ARP information and gather thousands of these packets very, very quickly.

Once you have all of these initialization vector packets, it only takes a matter of seconds then to extrapolate the WEP key from this. And of course, once you have the WEP key you have access to all of the data going across that wireless network. You can see this cryptographic vulnerability in WEP was significant and it was made so much easier to take advantage of by using this **ARP** replay attack.

**Tags:** [arp](#), [certification](#), [comptia](#), [replay](#), [security](#), [wep](#), [wireless](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **WPA Attacks – CompTIA Security+ SY0-401: 3.4**

An encryption technology doesn't have to suffer from a cryptographic flaw to be susceptible to attack. In this video, you'll learn how an encryption protocol as strong as WPA can be successfully hacked.

When the cryptographic problems were found with the **WEP encryption protocol** it was a big issue. Something that needed to be resolved. And of course, we went back to the drawing board and came up with a new type of encryption called **WPA**.

**WPA** itself was one that was a little bit more secure. We didn't have that same problem with the initialization vectors and the replay attacks that we had with the **WEP encryption protocol**. There were eventually some various minor vulnerabilities found with the WPA protocol surrounding the **TKIP**, but these were very specific cases.

The devices had to be in a particular place for this to work properly. There was a man in the middle. It was not completely obvious. But nonetheless, it was still a vulnerability and something that people had to be concerned about.

When we released **WPA2**, it was a protocol that worked differently. **TKIP** was replaced with **CCMP and AES**, which were very strong protocols to be able to encrypt information. And there are even to this day no known cryptographic vulnerabilities in WPA when you're using that **CCMP technology**.

Without any known cryptographic vulnerabilities then we have to use other methods to be able to crack or hack into a **WPA2 network**. If you're running WPA2 on your home network then you're probably using **WPA-Personal**. You might also see this referred to as **WPA-PSK**.

That stands for pre-shared key. That means everybody on the network has the same key, and we all use the same key on every system to gain access to the wireless network. That means the only way you can really get in is to find out what that key is. And you could perform a brute force or a dictionary attack to try to determine what that key happens to be.

There's no other way to reverse engineer that key out using a cryptographic vulnerability because currently we don't have one within WPA2. That means on your wireless network, you would do best to create a relatively complex key. It needs to have a lot of different letters and numbers. It needs to be as long as possible. And if you can, avoid any obvious words that somebody may be able to run across with a dictionary attack.

If you're in a larger environment, you're probably using **WPA-Enterprise**. You may see this also referred to as **WPA-802.1X**. That is the network access control mechanism that would be used to authenticate people onto a network.

So everybody uses a different authentication method to gain access. You don't just hand out a shared key to every one, which is not something usable in an enterprise that way if somebody leaves the organization you don't have to change all of your keys. You would simply login with your username and your password and that's what gains you access to the wireless network, usually with some type of authentication mechanism on the back end, like **RADIUS**.

And again, there's no practical attacks against that either. You would simply need to determine someone's username and password. In fact, this would be harder to brute force because now you only have access to brute force in individual user's username and password and not simply a global key that everyone is using. So if you're planning to get

into a WPA2 network you need to understand that it's going to be extremely difficult because it's effectively a brute force or a dictionary attack to gain that access.

**Tags:** [certification](#), [comptia](#), [security](#), [wireless](#), [wpa](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **WPS Attacks – CompTIA Security+ SY0-401: 3.4**

If you can't break the encryption, maybe you can break down the door. In this video, you'll learn about a significant security flaw in wireless devices that use WPS.

When we're working with security on wireless networks we often refer to the WEP encryption protocol, or the WPA encryption protocol, or WPA2. But there's a mechanism on a wireless access point called WPS. This is not an encryption protocol at all. It stands for Wi-Fi Protected Setup. It is something that used to be called **Wi-Fi Simple Config**, and as the name implies, this was a mechanism added to wireless access points that was to make it much easier for devices to securely connect to these access points. Adding a WPA2 key, and then distributing that key to all of your devices, seemed to be a little bit too complex for the novice. So we created this very simple method using a PIN, using a simple series of numbers on the wireless access point that would then allow us to easily connect our devices onto the wireless network.

There were a number of different ways you can connect to the wireless network. You would have a **PIN configured** on the access point. It's usually written on the access point somewhere. And then you would enter that PIN on your mobile device. You could also bring your mobile device nearby and push a button on the wireless access point that would then allow the remote device to have access. Some cases, the wireless access point took advantage of NFC, Near-Field Communication. All you had to do was get your mobile device close to the access point and it would then allow you access to the wireless network. There was also a USB method that was used. It's no longer something that applies to the **WPS standard**. But on some older access point you may still see a reference to a **USB connection**.

Although the idea of WPS was a good one, unfortunately, it was the implementation of WPS that ended up being its undoing. And in December of 2011, there was the discovery that there was indeed a design flaw in WPS. And it's a design flaw that's been there from the very beginning. The WPS PIN, if you look at it, it's an **eight-digit number**. It's really seven digits and a checksum. So if you needed to brute force these seven digits to try to force your way on to one of these wireless networks, you would need to go through about 10,000,000 possible combinations. Well, that seems secure enough, doesn't it? The problem is that the WPS process validates the PIN in two forms— in the first half and the second half. So really it validates the first half, which is four digits, and the second half, because there's a checksum digit there, it's really only three digits that you would then need to validate.

That means to validate the first half was 10,000 possibilities and to validate the second half was only 1,000 possibilities. Well, these 11,000 possibilities is certainly a lot fewer than 10,000,000, which means if you wanted to run through every possibility for WPS, it only takes about four hours to go through every single one of them. And, obviously, if you're trying to gain access to a wireless network, you're usually somewhere where you're away from the network. You're somewhere where you can run through and do this and have it go through its four-hour process to find this.

And even worse, these wireless access points did not have a brute force lockout function. Which means you could go through all 11,000 of them, and all of them could be wrong except for the last one, and you would never be locked out of the process. This was

obviously an enormous security concern. And it's now recommended on everybody's access point that you disable the WPS functionality and don't use it at all.

**Tags:** [certification](#), [comptia](#), [security](#), [wireless](#), [wps](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Cross-Site Scripting – CompTIA Security+ SY0-401: 3.5**

Browser security flaws can allow information to be inappropriately shared between web sites. In this video, you'll learn about the different types of cross-site scripting and how to protect yourself against **XSS**.

One of the more prevalent and dangerous attack types is called cross-site scripting. We refer to this as **XSS**. There's something already called **cascading style sheets** that we use in our browser that is a technology called **CSS**. So to make sure there's no confusion, whenever you see cross-site scripting referred to we always use an **X** instead of a **C** for cross. This is a word or a term that was originally created because older browsers would allow information from one browser window to interact with the other browser window. And if those were on different sites, you could essentially take information from one website, or what you were in putting on one website, and send it to another. And if the bad guy could hide one of those browser screens and grab your user-name and password and send it to another site, well obviously, that is a security problem.

These days our browsers are much smarter than that. They've been programmed so that they will not allow a lot of cross-site communication. The bad guys are always trying to find ways to go about doing this but we have generically called now this methodology of scripting information and sending it some or else, we've now called that cross-site scripting. It's a very common vulnerability. It's also not very straightforward. It's something that you're really trying to take advantage of bad programming, of differences in how websites are developed. So some websites might be susceptible, others may not. It requires a lot of research and a lot of testing to see if your particular website is one that is susceptible to cross-site scripting or not.

You'll find that a lot of the malware out there is using JavaScript to do a lot of it's heavy lifting, and so cross-site scripting generally uses a lot of JavaScript. But as the name implies, it's very generic— it is scripting. And scripting could really be anything. But what we found is that **JavaScript** is so powerful in our browsers that the bad guys tend to gravitate towards JavaScript because it's able to do a lot of things for them. And if you have JavaScript turned on, you are probably susceptible to a large amount of cross-site scripting vulnerabilities and attacks that are out there. It's just one of those things where you have to keep your browser updated, you have to make sure your web apps are updated so that you are not susceptible to cross-site scripting.

There's two primary categories of cross-site scripting attacks. The first one we'll talk about, our **non-persistent cross-site scripting attacks**, you may also hear these referred to as **reflected cross-site scripting attacks**. These are attacks that are not part of a web page, these attacks are ones that are emailed to you or you are enticed to click a link that is going to run the script that's part of that attack. And it's trying to take advantage of vulnerabilities in user input on things like a search box, for instance. You'll get an email from the bad guy and they'll say, please click this link, here's a funny video for you, here's things you should look about. And that link is going to take you to a website that runs this cross-site script that then provides the bad guy with information that they can use to perhaps gain access to your account.

So they'll have you login to Facebook, they'll have you login to your financial account, and it's going to then— behind the scenes— send your credentials, or your session IDs, or some user information back to the bad guy. And then of course from their desk they now

have access to your account, they can do whatever they'd like. This is executing in the victim's browser. So they're essentially going to a website, and it's the browser that is now becoming the problem for us, it's running exactly the script the bad guy gave us and the browsers happily handing off our credentials to someone else.

That's not exactly what we'd like to have happen. When the bad guy has your session IDs, your cookies, some of that really important session information that they can use to get into your account, they can now do a lot of things and you have no idea that this happened. It all happen behind the scenes because that script told it to grab that session ID, send it off to the bad guy, and now they have access to your information.

Another type of cross-site scripting attack is the persistent attack or the stored cross-site scripting attack. This is one where the script itself is stored on the web server. The bad guy doesn't have to send information in an email that has the script inside of it, they're going to post a message on a social network or they're going to post a message in a forum somewhere and that message, as part of that message that's on the website, is going to have that particular piece of script inside of it that then attacks your computer. And now it is persistent. Everybody who goes to this website gets the payload. So it's a good way of affecting many, many, many people all at once. There's no specific target here, you're not emailing a specific person and trying to get a specific person's information, it's anybody who happens to go to that website who happens to have a browser that is susceptible to that particular cross-site scripting attack. And if this is a social networking site, obviously there are many people on these social networking sites, and these forums it can spread very, very quickly it can be propagated very, very quickly.

And we've seen some very good examples of this in the past. In October the fourth of 2005, a gentleman named Samy Kamkar realized that he found a way in Myspace to force people to become his friend. So he thought this would be a great way to build his friends list. You can have a lot of friends because you can force people now to become your friends. So he wrote a little cross-site scripting information and he posted it online. So this is a persistent or stored attack that he had on Myspace. And what it did was post a message that said, "but most of all Samy is my hero" on the profile of the victim and then it added them as a friend to Samy. And so suddenly, once it added as a friend, it posted the persistent script to their own profile, it essentially became a worm.

Everybody who saw it ended up having that script copied to their profile, their friends saw that script, all of them got infected with the same thing and so on and so on. 20 hours later Sammy had over a million friends as part of his Myspace profile, and unfortunately he also had a felony. So part of the problem with this is that he was manipulating the system. It was a hacking attempt. It caused a lot of problems for Myspace because this worm was out of control. He essentially brought down the Myspace service.

He ended up having a plea agreement to a felony. He got three years probation, 90 days community service, restitution that he had to make to Myspace he actually had the ability to use a computer taken away from him— he could not use a computer or network device at all. Eventually that was given back to him. He had that particular piece of it erased from the record. And now he is a security researcher. You can go out to YouTube you can see some of the things he's done with entrepreneurship. So now Samy is making our networks safer and using this knowledge that unfortunately got out on Myspace, he's now using it for the greater good.

Let's look at a practical example of how somebody could use this cross-site scripting capability to gain information they would normally not have access to. Let's take this example right in front of us. This is something called **WebGoat**. This is from **OWASP**, this is the **open web application security project**. You can go on to [www.owasp.org](http://www.owasp.org) and you can download this very vulnerable web front-end called **WebGoat**. It's a series of examples of how you can try to take advantage of some of these flaws. It's essentially a

badly programmed website. And you can test and see how some of these vulnerabilities are affected on a website and try some different things yourself.

So let's try a cross-site scripting problem ourselves. Let's say that we are an employee. Let's say that we are, if I look at this HR department application, let's say that I am Tom Cat and I'm going to login with my credentials. And in my credentials, if I scroll down a bit here, you can see that I have the ability to look at my profile. And my profile for HR has my name, has my street address, has other information inside of it. But I want to gain access to everybody's profile and I know that the person in charge of the HR department does have access to everybody's profile. And inside of your profile you can see people's salary, what they make, you can get credit card information. So having that particular HR account might gain us information that we normally would not have access to, and probably shouldn't have access to.

Well, what we're going to do is embed a cross-site scripting attack right in our own profile. Maybe I'm not even Tom, maybe I've just gained access to Tom's profile. I'll use his profile as a jumping off point. So I'm going to edit this profile and in the street address here I'm going to add some additional text here and I'm going to put in here and create a JavaScript. I'm going to write here a script, and I'm going to inside the script do an alert, and inside the alert I'm going to put a session ID, and inside of that I'm also going to ask this to add on the cookie information. And in document, cookie. And inside of JavaScript—that's the way that you would add on some additional cookie information— and I'm going to in the script right there. So I've essentially added on a little bit of JavaScript inside one of these fields in this particular form. It probably should not be working, this form should not allow me to do that. But I'm going to update this profile. In fact when this profile runs, you can see the session ID information is right here.

We have a later video that talks about session hijacking and how that particular piece of information is a very, very valuable piece of information. Well that's good, now I've embedded it inside of my profile. So I'm going to log out. Now I'm going to send a message to the guy in charge of the HR department and say, could you look at my profile? I think there may be a problem with it. Please have a look, I think there's an issue. Well then if it's coming from Tom, who may be employee or it appears to be coming from Tom who may be an employee, that's something that we really may be interested in making sure that his profile is OK. So we'll login as Jerry. I don't think I've got the right login name there.

So we're now logged in as the HR guy. Notice the HR guy has the ability to look at Tom Cat, Jerry Mouse and Joanne McDougal's information. So I have more information available to me. But Tom sent me an email that said he's having problems with his profile, let's click on Tom's name. I'm going to view the profile. That script runs and it runs as the person in charge of HR, it runs as Jerry. The session ID is displayed, if I was a bad guy instead of displaying that session ID, I would send it to me.

And if I have the session ID of who's logged in right now, I can essentially pretend to be them directly to the web server. The web server uses the session ID as a piece of information that determines who's logged in. So I wouldn't need Jerry's user-name and password at that point to hijack his session and be able to see all of the information that he might have access to. All by planting a little bit of script inside of Tom's account now I can see everything that Jerry can see.

To protect against cross-site scripting there's a few things we should just always keep in mind. One is, don't click links inside of your email. I say that over and over and it's something that pretty much everybody should think about. There's usually going to be a link in your email, even if the link is legitimate it should just be a best practice for you to take that information, go to a browser, type it in manually, and go to the site that you need to go to. You might want to also consider disabling JavaScript or having it only run on

certain sites that you might trust. There's usually extensions, or add-ons you can get for browsers, that what a allow you to do that on a per page basis or a per site basis.

You should also make sure that your browser's updated. If you're in charge of a web server, in charge of applications on that web server, you need to also make sure that all of those applications are up to date because manufacturers and developers will find new ways that people are taking advantage of these scripts and they'll patch them. So by having the latest version of that on your web server you might be able to avoid it. If that HR department had the latest version of that HR software, it may have already stripped out any of that scripting and they wouldn't be susceptible to that problem. By keeping your web apps up to date, and keeping your browser up to date, and the applications, and the operating system on your computer, you can be assured that at least those known cross-site scripting attacks are ones that you can prevent and make sure that nobody takes advantage of your browser.

**Tags:** [certification](#), [comptia](#), [cross-site scripting](#), [security](#), [XSS](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **SQL Injection, XML Injection, and LDAP Injection – CompTIA Security+ SY0-401: 3.5**

Database security can sometimes be circumvented by poorly designed software. In this video, you'll learn how **SQL injection**, **XML injection**, and **LDAP injection** can get data from the most secure databases.

**Code injection** is a type of attack where you're taking your own code into your own exploit and your embedding it within an existing data stream. You may be able to do that from a website. You may be able to do that by manipulating packets as they're going by. There's many different tools to be able to do that. And there's many different types of code that you can use and inject into a stream of data. This is usually enabled or it's something that is available as an exploit because somebody's done a bad job of coding the program. Normally applications should look at what people are using as input and clean it up and make sure that people don't take advantage of these injection type vulnerabilities. But if you're doing filtering and you're properly handling the input and the output from an application, you shouldn't be able to inject your own type of information into the middle of that.

There's so many different data types. People do data injection of **HTML**, of **SQL traffic**, **XML**, **LDAP**. There's many different ways to put your own little piece of information within a stream to try to get around some of the security that might be there. When we think about large enterprise databases, we usually think about **SQL based databases**. This is **SQL**. The stands for **structured query language**. This is a very common language that many of the largest databases out there will use. You'll see this in relational database management systems. It's a very, very common database language and it's very powerful database language, which is why we see it in some of these very, very large databases.

Well, one of the things you can do is inject your own SQL code into some of these streams and you end up having the **SQL injection**. You're modifying the actual SQL requests that are being made to a database. And you're doing it through a web front end. Usually you don't have direct access to the database. You talk to a web server behind the scenes, the web server then talks to the **SQL database**. But if I can give the web browser in front of me bad information to give to the web server, which then gives bad information to the SQL Server, I can then get information out of the database that perhaps the developer never intended me to be able to do.

If you look at SQL injection it's very prevalent, but there's other types of injection as well. For instance **XML injection**. **XML** stands for **extensible markup language**. It's a very common format that's used these days to transfer information between point A and point

B. We can store information, transfer information, and it's a standard format. Because it's a standard format there's an opportunity here to inject my own type of XML requests inside of that to be able to change things however I would like to change them on that web server. Good applications will validate this XML, badly or poorly programmed applications will not. And that's an opportunity for the bad guy to get into this XML to be able to have your application do things that perhaps you were not intending that application to do.

Another type of code that susceptible to injection, we see this often, is **LDAP**. **LDAP** stands for **lightweight directory access protocol**. It was actually protocol created a long time ago by the telephone companies, they need an easy way to access user names, to access your first name, last name, your address, and your phone number, and it needed to be in a massive database. And that's where, really, directory access protocol came from. What the lightweight directory access protocol is what we've now used on our computer systems these days and generally you see this as name services somewhere but other databases will also use this LDAP protocol as well. So if I can craft my own LDAP messages and insert them inside of an LDAP stream, I may be able to get around a poorly programmed application.

Let's perform some SQL injection that's going to allow me to login to an administrator's account without having the administrator's password. That obviously would be a pretty insecure system, and indeed this one we're using absolutely is. This is called WebGoat, these are a series of very, very vulnerable web applications. You can download WebGoat from the open web application security project that's at [www.owasp.org](http://www.owasp.org)

This is a human resources app and what I want to do is not login as an employee, I want to login as Neville, who is the administrator. But to be able to do that, I need to inject information into the data stream. And I'm going to use an add on in this Firefox that I'm using. It's an add on that you can download and try called Tamper Data. It allows me to modify information on the screen. Change the way this data is being seen on the screen. I'm going to make this window a little bit smaller so we can see what's going on.

What it allows me to do is as data goes to the web server I can stop it, modify the data, and then send it on its way. And we can actually do the reverse in the other direction if we wanted to. I'm going to turn on the tampering, I'm going to start to tamper. Which means any time I send information to a web server it's going to ask if I want to change it. So I'm not even going to type in a password for Neville, I'm just going to hit log on and tamper says, would you like to continue tampering? Will absolutely, yes, I would. Because as part of this information in the header and parameters that are sent to the web server in the password field I'm going to perform some SQL injection. I'm going to type in some SQL code here that is going to get around the very, very bad programming that is on this particular computer. So there's the SQL code that I have in place and when I click OK it's going to continue with the tampering.

And if you look, I am now logged in as Neville with full access to everybody's profiles. It is remarkable and quite powerful, this SQL injection. And that's why when the SQL injection problems occur, they're usually quite devastating because they provide so much access to the database and usually it's information that you don't want the bad guys to be able to see.

**Tags:** [certification](#), [comptia](#), [injection](#), [ldap](#), [security](#), [sql](#), [xml](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Directory Traversal and Command Injection – CompTIA Security+ SY0-401: 3.5**

The bad guys can take a poorly configured web server and get access to the server's entire file system. In this video, you'll learn about directory traversal and how to protect your web server against command injections.

A **web server** is a relatively closed environment. We get on to the server, we have access to certain pages and certain applications, and usually that's it. You don't have any more access than that. Of course behind the scenes there's a full blown computer, it's running an operating system, there's many files and many directories and that machine probably has access to other machines in your environment. But you're not supposed to have access to that. You're only supposed to have access to just the web server piece.

Of course there are times when that is not the case and that you do have access to other parts of the computer that you normally should not. And it's probably because there's a misconfigured web server, or the web server itself has some vulnerable code, or maybe the applications you're using on that web server are giving you access to that particular web server computer that normally you shouldn't have. This is generally a bad thing because if the bad guys get a hold of that, they can run their own programs right there in the web server if they want to. And if they're using directory traversal to get around what normally would be your website, they can run the programs that are on your computer that normally they should not have access to.

When we're looking at the directories of a web server it look something like this. This is one that I'm running right now where right off the root I have a lot of different folders here. My web server folder is actually under the lamp P HT docs directory and inside of that is the web server. This is really, in a green, the only part you would see if you went to the web server. You're not supposed to be able to go to any of these other directories back here. You're only supposed to see this little world.

So what happens if you get access to some of these folders and you can run applications there and that command injection? Let's try doing some of that ourselves and see what we can do. Let's see what happens when we use a misconfigured web server and some vulnerable applications to get around some of the directories, to be able to traverse those and to inject our own commands into these applications.

I'm using this application called **DVWA**. You can go to Google and search for the **DVWA**. It's a vulnerable web application set of test that you can do. So you could run this on your own servers and do the same thing that I'm doing here. There's a set of file inclusions here that I could run and where it gives me information on the screen it just outputs what's in a file. Well, I'm going to use this very poorly worded script to output a file that's located elsewhere on the server. So what I'm going to do is get rid of this Include **PHP** that's being included. I'm instead going to want to go to the **ETC** directory and simply write out the password file and hit enter. And you can see, if you're familiar with the **Linux Unix** password file, this is the file. So I've been able to take this bad app or this bad script that runs on this web server, go into a section of the web server I should not have access to, and output a file that has sensitive information about what's on this server.

That's just one way of traversing those directories to be able to see that information. Now let's inject our own commands into this. There's another one of these that is a command execution that normally you'd be able to ping from here. So I can put in an IP address and hit submit and it will ping out there and give you the results of that ping command. And there it is on my server. There's the results. But it's a portly written application and I can escape around some of these things and put my own commands in there, like the ifconfig command to see what the configuration is of this server. So anything the web server can run, any applications that can be run by that web server, I'm able to run right

here in the web browser. Obviously, having that directory traversal and being able to inject my own commands into this puts it this particular server at risk. And those are the things that we'd like to try to avoid when we're configuring our web servers and our applications.

**Tags:** [certification](#), [command](#), [comptia](#), [directory traversal](#), [injection](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Buffer Overflows and Integer Overflows – CompTIA Security+ SY0-401: 3.5**

A poorly developed application can allow the bad guys to manipulate memory using buffer overflows and integer overflows. In this video, you'll learn how an overflow works and what you can do to prevent these security issues.

An integer is a whole number. So the number seven is an integer. The number 7.1 would not be an integer because it has a decimal place or a fraction associated with it. Negative numbers can also be integers. So 32 and negative 32 are both integers. Usually there is a fixed boundary when we're developing applications for these integers and you assign a variable to cover the range between, say, negative 32 and 32. So every possible number between those would be an integer that we could store for that particular variable. The problem is if we don't do a proper job of developing that software, that are software may allow that number to go a bit out of bounds.

But since it's such a restrictive area, once you go from 30 to 31 to 32 instead of going up to 33— which is outside the scope for that variable— it's possible that you roll all the way to the other end of the range. So you could go from 32 to negative 32 to negative 31 and work your way back up that way. This is a bit of a problem when you're developing software because you don't want these variables to be something unexpected.

For instance, if you were allocating a space in memory— you were creating a buffer area to store some other information and the size of that area was based on this integer, and we were expecting the integer to be 32. But when we're ready to allocate the space, instead of having a 32 as represented as that integer, it's now a negative 32. And obviously we can't allocate a negative amount of space in memory, and our application would fail. These are the types of vulnerabilities that the bad guys are looking for. They know if they can overflow that variable and make the integer for buffer allocation be negative that they can cause that application to fail. And this is a perfect opportunity to create a denial of service for this application.

A buffer overflow is a different kind of overflow. Buffers are areas that are allocated in memory and we put information into those areas. Before anything can be written into memory, the developer has to carve out a little space inside of the memory of your computer and that's where they're going to store a lot of the variables and information that they need during the execution of that application. The developers, though, need to be very careful about how they check what you're putting into that buffer. You have a buffer set to a certain size, you don't want to allow someone to store something larger into that buffer or you'll have a situation where you have a buffer overflow. And buffer overflows can be very powerful for the bad guys when used in a very particular way.

But it's really difficult to find that particular situation where you can overflow a buffer, do it in a way that you can always expect a certain outcome, and avoid crashing the computer. Instead have the computer provide you with something special like root access or administrator access to the operating system. So if you can repeat that buffer overflow over and over, then every system that is using that application or that operating system that is susceptible to the overflow will now be accessible to the bad guys, as well. And that's what we're trying to avoid we try to patch or applications that might be susceptible to a buffer overflow.

This is a visual representation of what a buffer overflow might look like in a very simplified way. But it does speak to how you were able to manipulate information by taking advantage of a buffer overflow. In memory we have a variable A and a variable B. And currently variable A has not been allocated anything. No one has put information into that buffer. It's got a number of bytes available but nothing has been added to that particular area in memory yet. Variable B, however, does have information in memory. It has the number 1979 in decimal– which hexadecimal is zero seven BB.

The bad guys have determined that if they can change variable B, then they might have additional access to your computer. What they're going to try to do is change this byte, which is hexadecimal zero seven, they want to change that to something else. So what they'll do is add information into variable A that is going to overflow into variable B. And what we'll do is we'll add a value of excessive. This is e- x- c- e- s- s- i- v- e and in the end would normally be an e.

You can see this overflows into the next byte. And if you look at the hexadecimal value of B now, it's changed because the number 65 is in the front of it. So six five zero zero in decimal is 25,856, which is a very different number than was there originally. All we did was change what was stored in A and we somehow were able to modify what was stored for B and that's not supposed to happen. That's your buffer overflow, and at this point the bad guy may be manipulating variables to do whatever they'd like to on your system.

**Tags:** [buffer](#), [certification](#), [comptia](#), [integer](#), [overflow](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Zero-Day Attacks – CompTIA Security+ SY0-401: 3.5**

Many of our applications contain security vulnerabilities that haven't been discovered yet. In this video, you'll learn about zero-day attacks and why it's important to patch our systems as quickly as possible.

Many of the applications that we use every day have vulnerabilities. Many of the operating systems that we currently are using have vulnerabilities. But we just have not found those vulnerabilities yet. Security researchers are working all the time to try to find where there might be a program that's not working the way it should be or to find a hole in an operating system that they can exploit to be able to gain access to that system. Now the good guys are looking to find that vulnerability and if they find it they'll generally call the developer or the manufacturer of that operating system and tell them that they have found a vulnerability. And then it's up to the people who've created the application– those developers or the developers of the operating system– to be able to create a the patch that will close that hole.

Now in the meantime of course, the vulnerability still exists. And that's what the bad guys are trying to find. They're trying to find a vulnerability that nobody else has found yet and that the developer has not patched. And if they find one of those they can do a lot with that operating system. They can get malware onto your computer much easier, they can take over your system, they can put a botnet on your computer. They can really take advantage of that for their personal gain and that's what they are after. And when we find these situations where the developer has not patched a problem or perhaps this vulnerability has been discovered by someone else and we see it in the wild and it's open and everybody is susceptible, we call that a zero-day vulnerability. Which means the vulnerability has either not been detected or the vulnerability itself is not something that has been published for everyone to know about.

So in other words, no one knows this vulnerability is there, it's a zero-day vulnerability. You often see zero-day exploits that you're getting a message from a manufacturer of an operating system, you're getting a message in an email as a registered user of an

application saying, we've just identified this problem, it's something we've even seen in the wild that people are actively taking advantage of and here are some things you can do to help mitigate this problem in the meantime until we come out with a patch. So often you'll hear about a problem, but you won't even be able to patch it until a number of days or even weeks has gone by.

And that is especially bad because then if all the bad guys hear about, oh there's a vulnerability out there, they're going to try to figure it out. Because the developers not going to tell you, here's how you take advantage of our operating system, here's how you take advantage of our application, they're just going to let everybody know you need to be aware there is a problem. And that puts the other bad guys on high alert. Because if somebody's found a hole, they're going to try to find that hole as well.

A good place to go to look at these particular vulnerabilities, especially zero-day and other kind of vulnerabilities, is the common vulnerabilities and exposures website you go to [cve.mitre.org](https://cve.mitre.org) and you can see what all of the latest vulnerabilities are, both the zero-day and those that are not zero-day vulnerabilities. And it's also good reference to go back in time to see what's happened in the past. Now if you're trying to look at an operating system or an application trying to determine should I be patching this, are there things that I need to apply to this app to make it more secure, that's a good resource to use.

Let's look at the scope of a zero-day vulnerability. And let's go back in time to November the third of 2010, Microsoft announced that there was a zero-day exploit for Internet Explorer 6, Internet Explorer 7, and Internet Explorer 8. So this was a pretty broad and very large impacting zero-day vulnerability because it affected practically every used version of Internet Explorer going back a number of years. They released a security advisory that just had basic information because they don't want to give away too much information about how the bad guys were exploiting Internet Explorer, they just wanted to make everyone aware of the problem and things they can do to make sure the problem would not affect them.

On December the 14th, so well over a month after the announcement, the patch finally came out. So MS10-090, the patch was released and it talked about the vulnerability that was related to vectors in cascading style sheet token sequences, the clip attribute, an invalid flag reference. A lot of details there. And at that point Microsoft said here's where the problem really was, but they were telling us this detail because they had a patch. And it was up to you, the end user, to make sure those patches were put on your computer. Obviously if you had not patched your computer, Internet Explorer 6, 7, and 8 would still be vulnerable. And probably to this day you still have people that have vulnerable Internet Explorer versions for this particular vulnerability and the bad guys absolutely want to try to take advantage of that.

But look at the scope here. **Internet Explorer 6** was released in **August 27, 2001**, so this particular vulnerability was just sitting there. Nobody had found it yet. And only until the bad guys found it and it was announced on November 3, which was just over nine years gone by since that version have been released. So it had been sitting there. And this is what is the problem, the fear for our security professionals, is that we know the applications have vulnerabilities. We know there are things that we don't know about yet and that's our biggest concern.

If the bad guys find the vulnerability first they're going to take advantage of it. And if there's not even a patch available you are wide open for that vulnerability to take effect. And that's why we talk about layering different types of security on top of each other so that if an application does happen to be insecure or have a vulnerability maybe there's an **IDS** that can detect something interesting going on. Maybe you've set different things in your firewall to prevent access to other parts of that app. Those are the things that we do to help mitigate these zero-days. And we absolutely want to stay on top of exactly what our manufacturers for our operating systems, our manufacturers and developers of our

applications, are doing to make sure that we aren't affected by these zero-day vulnerabilities.

**Tags:** attack, certification, comptia, security, vulnerability, zero-day

**Category:** CompTIA Security+ SY0-401

### **Cookies, Header Manipulation, and Session Hijacking – CompTIA Security+ SY0-401: 3.5**

If you have the right information, it may be possible to gain access to a user's account information without any authentication. In this video, you'll learn about session hijacking and I'll demonstrate a live session hijack by gaining access to cookie information and manipulating **HTTP headers**.

You've probably heard of browser cookies before. This is little information that the browser stores on your hard drive. And it does this so that later on, if you were to close your browser and go back into your browser, your information would still be in there. Maybe it's information that allows you to easily log into a website. Maybe it's information that keeps your session active so that you don't have to log into a website. They're not generally considered a security risk because they are not executables. But the information inside of the cookie can be a security concern if somebody happens to get their hands on that information. Sometimes the developers of a website will store session information or other details that could be considered very sensitive information and would allow someone else to gain access to your sessions. We're going to look at this in this video.

So unless somebody is information or access to that information, it's not really that huge of a security risk but it does very often store very detailed personal information, private information, and sometimes that's more of a privacy risk than a security risk. So that's something you should also consider is that your name's, your email addresses, session IDs, and other information are being stored on your computer. And some of these cookies are used to track what websites you go to and keep information that it then sends back to a central database over time. That may be a security or at least a privacy concern, as well.

The real key for what we're going to look at today is that some websites, many websites like Facebook, like eBay, and many, many others will store the session ID in the cookie. If you've ever closed out your browser and you've opened your browser back and you've been able to log into a website without having to type in your user-name and password, it may be that the session ID itself is one that is persistent. It stays there. It allows you to connect over and over again without having to re-login all the time. Sometimes it's one where you're just keeping your web browser active.

You have different controls over the cookie if you're a developer, you can decide whether that cookie is only available for the lifetime of the browser, whether it's only available for a certain amount of time, maybe it's one that stays regardless of whether you close the browser or not. So you never are quite sure. You have to go to each individual cookie and look if that's something you're interested in seeing.

The way that this works is you log into a website— you log into Facebook, you log into eBay, you log into almost any website these days— you say, hi I'm logging in, here's my user name and password. And if you are authenticated properly, the web server sends you back a session ID, and it probably sends back a number of other pieces of information that are stored in the cookie, but we'll look at this very high level. It sends back a session ID. And here's this big hexadecimal session ID that's sent back. And as long as you send and receive information back to that web server— you send an email, you send some

data— and you include that session ID, both the web server and your computer are in sync and it recognizes, oh, you've already logged in. I've already given you a session ID, great, I'll allow you to send that message. I'll allow you to give me that piece of information.

But if the bad guy happens to get that session ID, they can sit now anywhere on your network or even not on your network— anywhere, really— and send this information back to the web server and say, oh, I'm actually this machine and here's my session ID which happens to match the session ID up there. The web server has no idea of this difference. It has no idea that these are two different machines. It assumes that the session ID is the same, therefore, it must have been the machine that originally authenticated. Now obviously, it's not a simple process to get a session ID, it's not a trivial thing to be able to use that session ID, but there are a number of easily available programs allow you to together this information.

You may recall in our video on cross-site scripting that was another one that we used to be able to grab that information and display it on the screen, or give it to a bad guy. If you're on a network where there are other users you can use things like Wireshark or Kismet, especially for wireless networks, to be able to gather information. These session IDs are usually sent in the clear back and forth over the network. So if we're able to grab the packets, we can easily see that cookie when it's transferred back and forth.

**Cross-site scripting** as we already mentioned is another good way to get that session ID and to be able to exploit that. So cross-site scripting is something that we always have to be careful of. Then we want to modify the headers that we send in so that we can pretend that we are that user with that session ID. So you may use programs like Tamper, or Firesheep, or Scapy that have either automated or very manual ways to modify the headers. What we're going to do in our case is use an add on to Firefox called Cookies Manager Plus. And that's going to allow us to have a very easy front end to be able to modify and add cookies into our browser.

To demonstrate this session ID hijacking and using this cookie manipulation to be able to take over someone's session I've got two machines running on my desktop at one time. I've got this **Ubuntu system** and this one is running Firefox. And I'm connected to this device and I'm able to load up and show you information here. Also I've got exactly the same thing running to the same server, the same web server, and this is my local computer. So I've got Firefox running on my local machine and Firefox running on a separate computer.

So we're going to login on the Ubuntu system I have here and we're going to look at the session information. Now this happens to be an application that was not well developed, it allows me to do this session manipulation to it. This is WebGoat. This comes from the open web application security project. You can download WebGoat at [owasp.org](http://owasp.org) and you can load it on your computer and try this exact same system. Also in Firefox under Tools you can see that I have already added Cookies Manager. You would go to your add-ons administrator, your add-ons Manager, and load up Cookies Plus Manager yourself.

So what we'll do is login. We're a normal user. In fact this particular WebGoat tells you to login using WebGoat WebGoat to see what happens. And let's do that. Let's login with WebGoat with password WebGoat. And this is a very simple application. It doesn't do anything other than tell you that you're logged in. So you can hit refresh a few times, resend this information. We're still logged in. So nothing really unusual about that. Nothing changed with this. If we go over to our other computer and refresh, you can see we're definitely not logged in to that computer. Now because we're logged in as WebGoat, we should be able look at this cookie information. So under my Tools pull down menu, I'm going to go to my Cookies Manager. And you can see I've already got a filter here for this single computer.

So we're just looking at cookies for that IP address. And I have a session ID. You can see the session ID content right here, this is the value of the session ID information. And here is the auth cookie, and just the name of auth cookie makes us realize this is the cookie we got when we were authenticated. So there are some things we can try to be able to take advantage of that auth cookie. If all we need is that auth cookie, maybe we can re-create this on another computer. So all the bad guy needs to get is some of this information that's within the cookie. So let's try that. I'm going to highlight this, I'm going to copy that auth cookie. And what I'd like to do is re-create it on my computer over here. So we're going to move to the other browser and we're going to use that auth cookie. Let's pull this down and I'm going to pull up my Menu. Under Tools we're going to choose our Cookies Manager as well. And here's the Cookies Manager for this computer.

Notice I don't have an auth cookie on this computer, we need to add that new one. So let's do that. And let's type in exactly as it was on the other auth cookie. And for the content let's put in exactly that same auth cookie. I'm just going to paste it right in. And click Save. Now let's see what happens here if I do a refresh. Notice because I have exactly the same auth cookie I'm now logged in. I didn't have to put in a user-name. I didn't have to put in a password. I'm simply now logged into this computer, exactly the same way the other user was. And this application is really bad because I can even log out, you can see clearly I'm not logged in anymore. I can refresh this page, I'm not logged in. But if I refresh over here, I'm still not logged out.

So even though the other user turned off their computer— they went home, the cookie is gone in that other machine— because I was able to steal it, I'm still authenticated, and I'm still in this application, and I can still use it exactly the same way the real user would have been able to use it. And that's the way that the bad guys would take advantage of these victims is to be able to steal that cookie information, re-create it in their browser very, very easily, and now they have exactly the same rights as the victim.

Obviously you saw how easy it was to pretend that you are someone else on this network, and. It's something that is very easy to do with some of the most popular websites in the world it's a very common way to maintain sessions on these systems. So some of the ways that you can avoid this happening to you is to encrypt your session end to end. Do HTTPS all the way to the web server because the bad guys cannot grab your cookie information if they can't see it. If they're monitoring that wireless network, they'll see encrypted data. They'll have no idea what your authentication cookies, session cookies, or any of your cookie information might be.

Now this puts an additional load on the web server. Being able to do encryption is not a trivial task. It really involves a lot of additional resources. So not all web services will allow this. Not all websites allow this. But the ones that do, you can have your browser automatically connect in an encrypted form. You may want to try downloading some Firefox extensions. There's **HTTPS Everywhere** and **Force-TLS** are a couple that if you go to a website that will support encryption, it will automatically shift you to an encrypted mode just so you don't have to remember to use **HTTPS** when you go to those websites.

Another thing you might try to do is encrypt at least across the wireless network. Maybe it's from your computer to at least somewhere else. So that if somebody was in your local coffee shop on your wireless network that's in a hotel, they at least would still not be able to see things in the clear. You can see this for instance, using things like a personal VPN like **OpenVPN**, **VyperVPN**, or any of the other VPN solutions you might have at your office would ensure that nobody would be able to at least see that locally.

Other people might want to use things like session ID monitors. There's things called Blacksheep and Application-specific ID monitors for each individual app. That is not a perfect solution but it with the least allow you to see if yourself or other people are out there on the network. Blacksheep's an interesting one because it will send fake session ID information out and the bad guys never know which one of those session IDs is really

legitimate. Session ID hijacking, being able to manipulate these cookies, is obviously a security concern and you should be sure that the applications your users are using, and the ones that you're using in your environment are protected against these types of hijacks.

**Tags:** [browser](#), [certification](#), [comptia](#), [cookies](#), [header](#), [http](#), [security](#), [session hijack](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Locally Shared Objects and Flash Cookies – CompTIA Security+ SY0-401: 3.5**

A number of security concerns are related to cookies and the information that we're storing in our computer. In this video, you'll learn about Flash cookies and how these locally shared objects can potentially leak information to the bad guys.

If you're using Adobe Flash on your computer then you may have objects stored on your computer called **locally shared objects** or **LSOs**. Locally shared objects are also referred to as Flash cookies. And this is a place that the Flash player uses to store information on your computer. This is turned on by default and it's very common for applications to store information that they might want to use for later. This is a stored area that applies to all the browsers you might be using and any time you would use the Flash player on your computer. So everything is in one place, in one common directory.

Ideally, the LSO can only be read by the domain that added that information into your computer. So if you had for instance, example.com stored some data as a locally shared object, only example.com would be able to access that information. Thereby creating at least some level of privacy associated with that data. Although all of this data is in a shared directory, it is still only going to limit access to that information by the domain that originally stored it. For example, if you visit a website www.example.com and it's stored some information as an LSO, that information can only be read from Flash that is running on www.example.com. Example.com could pass that information off to another domain, but by default only the domain that created that information is allowed to view it.

If a **Flash program** is simply storing some local variables or information that it needs to operate, that is a pretty innocuous use of the LSO, but you can store anything as a Flash cookie. You can store browsing history. You can store information about where you are visiting and things that you've typed into your browser. All of that information can be stored as a Flash cookie. Many websites will use these flash cookies and they'll store that information but they may not directly tell you that they are storing this information as a Flash cookie. And what people have found is that some of their private information has been stored on their computer without their knowledge.

There have been a number of legal challenges associated with these LSO. Sometimes the private information that you think is on your computer and not available to anyone else could be made available to a third party and because of that a number of class action suits have been created. In some countries you have to be specifically told when these particular locally stored objects are going to be used, and you have to consent to it. If you visit a website for instance in the United Kingdom, you'll get a message on your screen that says, this website uses cookies, is that OK with you? And you have to then agree to use those locally stored objects. This is one of the latest challenges that we've run into regarding the use of our private information, all because of those local flash cookies that are stored on our computer.

**Tags:** [certification](#), [comptia](#), [flash cookie](#), [LSO](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Malicious Add-ons and Attachments – CompTIA Security+ SY0-401: 3.5**

We are constantly installing new software and add-ons, but how do we know if this software is safe to use? In this video, you'll learn about add-ons and attachments, and I'll give you some real-world examples of malicious add-ons.

One of our larger security concerns is not the things that are already on our computer, but the things we interactively add to our computer ourselves. We manually decide to add new software to our computer, and unfortunately, sometimes that software has vulnerabilities, it has threats inside of it, that then create problems for our system.

One common way to do this is with attachments. Email attachments are still being sent. There are a very, very common way to get a bad piece of data, a bad program on your computer, and infect you with malware or spyware or some kind of virus. That's because so many people have email, and there are so many different kinds of threats and vulnerabilities out there.

They're called attachments because they're attached to the email, and you just double click it, and you're able to run it on your computer. The bad guy's trying to force you to run that, to encourage you with social engineering, so that you'll be able to do that.

Almost always, attachments should always be considered a security risk. Even if you recognize who sent this to you, even if you recognize the file, even if you've already talked to the person about them sending the file to you, you should still consider that attachment a security problem, because you don't know if the other user might be infected with something that's latching on to the attachment and coming over to your computer.

You really have no way to know what's in there. So you want to be sure before you run anything that is an attachment that you know about it, you understand what's in there, you perhaps have scanned it with one or more antivirus or anti-malware packages, to make sure that it's absolutely safe.

Another way to get information onto your computer is through add-on's. Very good examples, with Firefox or Internet Explorer are browsers that we use every day. We can make them easier to use. We can add additional functionality just by adding some add-ons onto our systems, onto our browsers, makes it very, very simple to extend the functionality of our applications.

The problem, of course, is the bad guys know that we are going to install these add-ons, and very often we just assume the add-ons should be completely trusted. It's something that is in a big list of things. Obviously other people are downloading this and installing it. It's from a relatively trusted source. So why not just add on to our browser without even considering that the add-on itself might be a problem?

Here's the way these add-ons work in something like Firefox. This is my Firefox view. And under My Tools, you see I have a lot of add-ons that are already here, because you can see a lot of things in icons here that are not normal in a stock configuration of this.

If you go to the add-ons option here, brings up a completely different set of menus. I'm running Firefox 4.0, so mine looks a little bit different than previous versions, but look at this huge library of different add-ons. And I can search for any type of add-on here. I'm going to type, just for security a very generic name, and look at all of the different security add-ons I could put on my system. These all sound great. I'll load all of them up.

I'll put anything I'd like onto my computer. It's a relatively easy process to do this. If you would like to add, I don't even know what some of these, I'll just click install. And here's a good example of how it's quickly downloaded. I just clicked my mouse to get it there. I click Restart now, and now the add-on is installed in my Firefox.

And you can see under My Tools pull down menu, I can go, well, the add-ons has already come right back here, and now click here for more info about the websites. Now the add-on has already kicked in. It's already started gathering more information for me about that particular site. There's a new button that's appeared that gives me a reputation rating for that.

I don't know if this reputation pull down is something legitimate or not. I just clicked an add-on, and now it's in my browser. It looks legitimate, but you have no way of knowing. I may have put myself at risk just by adding that into my system.

So there's our trade off. We have a certain amount of trust we have to put in these add-ons, but they're obviously a situation where this could be risky to our computer. These worries are not completely unfounded. In February the 4th 2010, Mozilla, the developer of Firefox, found Trojans in two separate Firefox add-ons, and it's ones that you could just do exactly what we just did. Go out to the add-on site, click a button, and now it's installed on our computer.

Inside of that was malware. It was a Trojan. It was trying to get you to install that so it could put that on there. Nearly 5,000 downloads across two different Trojans. They weren't even the same Trojan. They were two different ones that they came across.

Now obviously they removed the add-on from the main library, so people could not download it anymore, but even if I went back to the add-ons inside of my browser and chose to uninstall that add-on, it was too late. The malware was already on my computer. Just by uninstalling the add-on meant nothing.

It did not remove the malware from our system. So obviously, these people that installed this particular add-on may have been infected, and most likely were infected, a number of them, by this particular problem.

This happened earlier as well. This was not the only time this occurred. There was a Vietnamese language pack that also had a Trojan inside of it that installed malware, bad piece of system of software onto the system, so this is something that happened before. So obviously, you have to think about these add-ons before you install them.

The problem in this particular case is that **Mozilla** was only using a single anti-malware engine to scan these add-ons. They were using **ClamAV**. So something they mentioned after the fact, after they had decided, you know what, we should probably scan this with more than one engine. So today they are doing multiple scans to multiple engines just to be able to be sure that when they put something inside of the library that it's something that is protected, that does not have something vulnerable inside of it.

But as you probably recognize as a security professional, just because you scan it with anti-malware, doesn't necessarily mean there isn't any malware inside. It could be a piece of malware that those **AV or anti-malware systems** knew nothing about.

So you just have to be very careful about what you're downloading, get a level of trust associated with it, and maybe after you've installed it also do your own scans and make sure your system is up to date with the latest patches and up to date with your latest antivirus and anti-malware signatures.

**Tags:** add-on, attachment, certification, comptia, malicious, security

**Category:** CompTIA Security+ SY0-401

## **Arbitrary and Remote Code Execution – CompTIA Security+ SY0-401: 3.5**

A serious programming error can open your computer to code that can be run from anywhere in the world. In this video, you'll learn about remote code execution and how you can avoid having your computer taken over by the bad guys.

When we run an application on our computer, we are executing code. Nothing happens on your computer unless you have some application, some program, running that is executing in memory. Now this executable code is a very specific kind of program. This is designed to perform certain actions on your computer. This is not the spreadsheet that you're using. This is not a word processing document. It's the program that you were using to edit a word processing document or the application you're using to manipulate numbers within the spreadsheet. So it's a very specific kind of file. And it may be related to a game that you're playing. Maybe it's a business application you're using. It could be something that is running as part of your operating system.

A number of executables run on our program behind the scenes just to make sure that all of the things that we're doing on our computer are working properly. Since nothing happens on our computer unless some code is executing, the bad guys really would like to have complete access to your computer to run whatever they'd like. Or they had like a program that you're already using to run some arbitrary code that they've somehow managed to get inside of that application. And if an application has not been developed well or it has a bug, then it may run some arbitrary code without any permission from you.

I mentioned earlier that there are a number of processes that are always running inside of your computer. And if any of those processes have a bug that allows for this arbitrary code execution, the bad guys could feed that arbitrary code to that process, it would then execute on your computer and then the bad guys would have whatever access they needed to your system. It is this original executable running as the process that created this problem to begin with, and normally you would be patching that process so that nobody would be able to run this arbitrary code.

We often think that a lot of these arbitrary code executions are something that can only happen if you have administrator access or root access to the operating system, but the reality is a number of programs can run in the normal user space and the bad guys just want to be able to start up their malware, get your system to run that as a normal user, and they'll be able to perform whatever functions they need on your machine. So all of us have to be very careful about these arbitrary code executions because even as a normal user you can really have the bad guys create a lot of havoc on your computer.

If you've ever looked through the release notes of the monthly Microsoft patches or you look through the notes associated with an Adobe patch update your sometimes run across a vulnerability identified as a remote code execution. These vulnerabilities are usually categorized at a very high severity because a remote code execution means that the bad guy can run software on your computer but they don't even have to run it or be anywhere near your computer. They can send information into your system remotely and have that execution occur on your system. This is obviously a significant vulnerability one that needs to be patched very quickly because you don't want people from anywhere in the world connecting to your computer and running whatever software they'd like.

**Tags:** arbitrary, certification, comptia, execution, remote code, security

**Category:** CompTIA Security+ SY0-401

### **Monitoring System Logs – CompTIA Security+ SY0-401: 3.6**

Your system logs contain a wealth of security details. In this video, you'll learn about the different log types and how they can be used to security your network.

In most organizations, we're collecting logs from every device that we have– the routers, the firewalls, the file servers themselves, and many other pieces of information. And this information can be very valuable for us to use not only for what we're doing internally, but also for making plans for the future. One of the challenges, of course, is that there are a lot of logs, so it takes some very specialized devices and technologies to be able to collect all of those logs, to parse through them, and store them, and then ultimately, to provide us with reports and information from what we've gathered over such a very large area.

You'll generally find different categorizations of logs, things like event logs, and auditing logs, and security logs. And each one of those log types provides us with a different kind of information that we could use for different scenarios. There's many options for automating the collection and the reporting of this log data. And you can find many open source and commercial packages that can collect all of this information, parse through it, and be able to provide you with some actionable data that you can use in your business.

An event log tells us any time something happens on the network. These are usually very normal operations. Someone logs into the network. Someone opens a file. A file is copied from one server to another. These types of situations are relatively innocuous. This is the normal operation of what's happening. We're simply logging every time one of these events occurs.

This might be useful to use, though, after the fact. If we're trying to determine, how did this file get transferred from one place to the other, we might have an event log that shows us exactly that information. Now, as it sounds, every time you store this information, you're collecting a larger and larger and larger log file. Event logs can be very large. And so you want to be sure that if you're planning to collect these, that you have plenty of storage set aside to collect as much information as you need.

You may be gathering these logs from many different places. They can come from your routers, in your firewalls, in your switches, in your servers. All of this is used to ultimately, after the fact, determine what happened on your network. And if you have a security event, these event logs will be very, very useful to help understand what happened before all of the alarms went off.

An audit log is very similar to an event log, but an audit log is only going to tell us when things change. And usually, these are things that are very important for us to be able to watch, so that later on, we can go back and see, who made that change, what type of change was it, what time of the day did that change occur? These audit logs might tell us when absolutely legitimate activity might be going on. If we're planning to make some firewall changes, the audit logs will be able to determine who made the changes and why they made them.

These audit logs can also tell us when unapproved activity has occurred. If suddenly, our log shows that a change was made to the firewall, yet nobody has any paperwork or any knowledge that any change was to be made, then you've got a problem. And that's where you may be able to find someone who's making unapproved changes, all from the information you're gathering from your audit logs.

You're not going to get quite as many audit logs as you have event logs. But in a way, your audit logs are almost more important, because we're looking for very specific

changes to occur in very specific places. And generally, these types of logs have very critical information inside of them.

As the name implies, an access log is going to tell us when somebody gains access to a resource. They may be gaining access to a file server. Perhaps they're logging in to use a VPN. There's going to be a log somewhere that tells us that that particular event occurred. This can come from web servers, which have their own set of access logs inside of them. There could be VPN concentrators. There could be applications that store log information when somebody logs into the application and gains access to certain types of data.

This could be very useful to tell who's gaining access, to make sure that people are getting the access they require from their resources. But it can also tell us who's not getting access to those resources. If somebody's constantly trying the same username but the wrong password to access a VPN, and they're doing it over and over and over again, you'll see that information in your access log. This way, you can start to limit the attack vectors available to you.

If somebody's trying to gain access to your web server and they're constantly trying to authenticate, and your access log shows that they've been denied access constantly over and over, you can create automation that might block that IP address, or limit access from that particular IP address, or slow down the process for them to make it very frustrating for what they're doing, or maybe you lock them out completely, so that nobody can log in with that particular username any longer.

If you're going back and trying to rebuild what happened during an attack, what did we see change, when did the bad guys gain access to a particular resource, all of that information is going to be inside of your access logs.

As a security professional, you're going to be looking through a lot of security logs. These are very focused logs, and they generally focus on very specific events that are occurring that are important from a security perspective. Usually, the file server team and the router team aren't necessarily interested in the security-related events. They may be more interested in the performance-related events.

So you'll find that you'll get security logs from all kinds of different places. They'll generally come from your security devices like your firewalls, or your VPN concentrators, or your IPS systems. These types of logs can tell us a lot about the security of our system, so it's very useful to be able to monitor these security logs over time.

There can sometimes be a completely separate logging system for the security team, especially since the information that the security team needs is so different than other parts of the organization. You might have a file server team that has a file server log collector. They're collecting performance information and availability for their file servers.

And you may instead create an entirely separate system just to gather security logs from those file servers. This not only is going to allow you to manage your own set of data, but it's going to allow you to find just the security pieces that are important to you and ensure that all of that data is being collected.

**Tags:** access, application, certification, comptia, event, logs, security

**Category:** CompTIA Security+ SY0-401

### Operating System Hardening – CompTIA Security+ SY0-401: 3.6

Out of the box, your operating system probably isn't the most secure. In this video, you'll learn some best practices for security your operating system from the bad guys.

When you first install an operating system, one of the things you commonly do before you ever connect it to a network or put it in production is to harden the operating system. And what we mean by that is to make your operating system one that is much more secure. We want to be sure we have all the latest patches.

We want to be sure all of our applications are up to date. There may be different tweaks we can do to make sure that our operating system is hardened. We're wanting to improve that security. But it's not just any one thing. It's a number of different things to be able to increase the security.

That's one of the challenges we have with making sure our operating systems and our computers in general are secure is that there are so many different ways the bad guys can get in. So obviously there's so many different ways that we need to consider when we're hardening up our operating systems. This also requires constant maintenance. We need to make sure that we have all of the latest patches.

Vulnerabilities are announced every month, sometimes in much more frequency. So we need to be sure that we have all the latest patches for our operating system and the latest patches for the applications that are running on that computer. In fact, we even, after putting the patches on and making sure our applications are up to date, may still find our systems vulnerable because of a configuration change that we made that allows the bad guys on to our system.

So we have to constantly monitor and maybe do checks of our systems to make sure that we haven't inadvertently configured our system in a way that's going to allow the bad guys to gain access. One way to harden our systems is to disable any services that may be running that we just don't need. They're unnecessary. This is a bit of a challenge. For instance, Windows XP, when you first install it, has almost 90 services installed.

Not all of them are active, but they are all installed. This is a screenshot of my Services view all my Windows XP. Some are started. Some are not. But you may not want to have all of them running, because all of those services may create an additional opening for the bad guys.

But we have to figure out which ones have the potential for trouble. The ones that we are worried about are the ones that have known vulnerabilities. But of course, we're also concerned about any services that have unknown vulnerabilities. But if we disable the service, we don't have to worry about the vulnerabilities. So there's the value in disabling some of these unnecessary services.

This may require a bit of research. These services are running **Windows, Linux machines, Unix space machines, Mac OS X**. Almost any operating system is going to have services that run in the background of some kind. So you may have to go the web, you may have to find out if this particular service on this particular server running these particular applications can be disabled without having any adverse effects.

Obviously, it's a bit complex. Sometimes it's trial and error. Let's turn it off and see what happens. Sometimes that's a good way to determine if it's a necessary service or not. So you test it, you monitor, and you make sure that once you've disabled those services that everything continues to run. And the smaller number of services you have running on a computer, the better the security posture is going to be.

Practically every network device you have out there has a management interface associated with it. Your firewalls, your routers, your switches, your **IDS, IPS systems**,

anything out there has a way to gain access, because your administrators need access to those systems. So obviously, you need the management front end. But what you don't want to allow is the bad guys to have access to this management front end.

These systems have a lot of sensitive data on them. They've got to have these interfaces available to us. So we want to do some simple things like set up some passwords so that if someone does gain access or finds the IP address that we would use to authenticate to this device, that they wouldn't have the right username or password to gain access. That's a very simplified way.

Most larger organizations are adding additional security on top of that. Not only do you have to be coming from a particular IP address or IP address range to gain access to that management interface, but it may require additional log-ins to get access to that interface. And it may require some third party authentication. We may have to have a token generation tool.

We may have these systems send us a text message with a number on it that we then have to type in to gain access. The only way that would work is if you were the one that had the token generator and you were the one was carrying that telephone.

All of our servers have accounts on them. They have usernames. They have passwords. So we also have to think about hardening up those usernames and passwords in those accounts. But weak passwords are very, very difficult to protect against. It's very easy for the bad guys to interactively try to log in as a particular user.

So very often, we have to have policies on our servers that if you log in more than five times with a bad password, we'll disable the account for a certain amount of time or perhaps disable it entirely. If they gain the entire database of those passwords, even though the passwords are not stored in plain text, they're stored as hashes generally, they can now offline do some brute force attacks against those hashes to try to determine what those passwords might be.

So we have to protect the store of those passwords as well. What we really want is our passwords to change often. And we need our passwords to be relatively complex. We want to be sure that if somebody gains access to those password files that at least the brute force attack would not generally be successful.

We're going to try to make sure or reduce the possibility that someone would not be able a brute force any of those passwords. This isn't always the best possible situation. Sometimes you're not in control of this. For instance, in December of 2009, the website rockyou.com had a security breach. It was a **SQL injection**.

And the bad guy stole 32 million account information. So they were able to gain access to usernames and passwords. And the problem with rockyou.com is that all of those username and passwords were not hashed. They were stored in the clear, as crazy as that sounds.

So that became very easy for the bad guys to gain access to this. But here's what's even more interesting. They posted the entire 32 million account database to the web. So now anybody could download the torrent, download the file directly, and be able to look through all 32 million accounts. It was obviously very embarrassing for rockyou.com, and it also created a number of security concerns for everyone else, because unfortunately, so many people use the same username and password wherever they might go on the web.

But what this did allow us to do is to perform some analysis of these usernames and passwords to see where are we having problems with these password policies that we have in place. A company called **Imperva** did a study of these passwords. And you can access the study itself at [imperva.com](http://imperva.com). Here's the download link for that PDF file.

What they found as they went through these millions and millions and millions of accounts is that 30% of the passwords were six characters or less. And in the rockyou.com case, the minimum requirement was five. So people were just barely going over the minimum requirement. 30% of them, six characters or less.

Obviously the smaller the number of letters in a password, the easier it is to do a brute force attack. 60% of the passwords were from a limited set of alphanumeric characters. There were no special characters in there. There was nothing that was outside the realm of what you would find a through z and zero through nine. And obviously, that makes it even easier for the bad guys to do a brute force attack.

50% of the passwords were names, slang, trivial things, or dictionary words. And in those cases, it becomes exceptionally easy to do a brute force attack. You want to do anything but this. You want to have very different passwords that are much stronger than words you might find in a dictionary.

The most common password— 123456. 290,000 of these users, the password was 123456. There were 79,000 users it was 12345. That's even worse. And of course, 76,000 users were— they made this one very hard— 123456789. 61,000 users had password as their password. Doesn't seem quite obvious does it.

And iloveyou came in, the last one here, 51,622. And of course, there's many more. They listed out the huge frequency in this download PDF file— very interesting. But it does point to how we have to be diligent about making sure that we have strong passwords and that we're auditing our systems to be sure that our passwords were strong.

It's very, very common for the security team to grab their password files and do their own brute force attack. How hard is it for us to find the usernames and passwords on these systems and determine are we really putting safe passwords, very difficult to find passwords on our systems? When we install an operating system, it usually installs a default set of accounts.

But we may not want all of those accounts to be on our system. After all, if we have an account, there is a possibility that someone could use that account to run programs or log into the computer. So if we were to disable or even remove accounts we don't want, we would then be making our systems just a little bit more secure and harden them up just a little bit more.

In the case of Windows, there's usually a guest account that's installed, even if it's not enabled by default. Linux or Unix has a number of accounts that can get installed or put into our system when we first install the operating system. But not all of these are necessary.

We may want to go through this list and start disabling or start removing some of those accounts. In fact, there's some of the accounts that maybe we just don't want any interactive logins whatsoever. You may need that account to be able to run different batch files or different processes on the computer. But you want to be sure that if somebody was to gain the username and password, that they would not be able to log in.

So that could be an additional piece that you can add on to help make your operating system just a little bit more secure and a little bit more hardened.

**Tags:** [certification](#), [comptia](#), [interface](#), [logs](#), [operating system](#), [password](#), [security](#), [services](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Physical Port Security – CompTIA Security+ SY0-401: 3.6**

Physical access to a network can be a significant security concern. In this video, you'll learn how to protect your physical network interfaces using configuration settings and access control software.

As security professionals, we spend a lot of time putting a lot of security on the border between the outside of our network and the inside of our network. But generally, we don't spend a lot of time securing everything on the inside. We have a lot of firewalls and intrusion prevention systems that are locking people out from getting inside. But we don't spend a lot of that money putting up those types of systems on the inside of our network.

The ports that are inside of our network therefore are generally wide open. We've got copper interfaces. We've got fiber interfaces. And of course, now we have wireless networks available. And so it's important that we secure also those physical ports that are inside of our network, just as securely as we secure the outside of our network. It's a constant balancing act to provide the right level of security on the inside of our network.

We want to prevent people from simply plugging into the network anywhere and gaining access to resources, but we also want to make the network accessible to everybody. There's a conference room. We want people to be able to plug in and perform their business function. We just want to keep out the bad guys from doing exactly the same thing.

One way to filter out the types of systems that can plug into your network is through something called **MAC filtering**. The stands for **Media Access Control**, and it's referring to the **MAC address** that is burned into the network cards that are inside of all of our computers. This would allow you to take an internal computer and connect it into that conference room. But if somebody brought their computer from outside of your network and tried to plug it in, that system would not have any access to your network.

This requires a little bit of work on the administration side. You have to collect all of the MAC addresses of all of your devices and you have to create a way to filter those out on every port that's inside of your network. One of the challenges with MAC address filtering is that these MAC addresses can often be spoofed. A lot of the software that we use as drivers for these cards allow us to put in our own MAC address.

And if we happen to know what a legitimate Mac address is, we can simply duplicate that MAC and now we have the same access that all of your computers have inside of your network. Many organizations will associate the access to the network with the authentication that you must provide. This is using a functionality called 802.1x, where your machine must first authenticate to a central authentication server. And only after that authentication has happened do you gain access to the network.

If anybody comes from the outside and simply plugs in, they won't have any access until they provide the correct username and password. Another good best practice is to administratively disable any switch ports that might not have anything directly connected. That way, you can be assured that nobody can walk into your closet, plug in from inside the infrastructure room, and then gain access to the network.

This also requires some additional administration, because you need to go through and make sure you know what ports are not physically in use and disable those. And then when you want those ports to be available, you obviously have to go back into your switch and administratively enable those ports so that they'll operate properly on your network. It's also a good idea to then do some periodic checks and make sure that nobody is using any ports they shouldn't be.

And if you've documented this switch configuration and you know what devices should be plugged in where, it should be very easy then to look at your switch and see very

quickly what devices may be plugged into ports that should not be in use. It may be very easy to find these unauthorized devices on a wired network.

But on a wireless network, it becomes a lot harder to find these rogue devices. You want to be able to perform audits and to be able to physically check your switches and to look at the lists of who might be connected to your wireless network. It's not uncommon to use network mapping software to be able to find everybody who might be connected to a network.

And then you can compare that list to who might be actually authorized to be on the network. It's also common to grab a spectrum analyzer, especially the portable ones you might use these days, and simply walk around your building to make sure that nobody has plugged in an access point that they brought from home, creating obviously a significant security concern on the wireless side.

And network access control can obviously provide you with a very secure method of authenticating people onto the network and only allowing the people who are authorized to gain access to your corporate resources.

**Tags:** 802.1x, access control, certification, comptia, mac filtering, security

**Category:** CompTIA Security+ SY0-401

### **Security Posture – CompTIA Security+ SY0-401: 3.6**

You can't build a security policy unless you know how to plan, monitor, and remediate security issues. In this video, you'll learn some best-practices for baselining, watching, and resolving security problems.

When you're building out a security posture, it has to be based on something. So one of the first things you'll do is build an initial baseline of what you would like your security to be. This often takes a lot of planning and a lot of thought. You have to look at the requirements that you have, the things that you need to protect.

There's generally a minimum level of protection you're thinking about for the data and the systems that you have in place. Windows systems, Linux systems, different databases, they may all have different requirements. And so you may be setting different baselines depending on the type of system. You also have to think about some of the legal requirements you have and some of the compliance requirements you have.

If you're a medical organization, there's a series of requirements that are defined in HIPAA that you must comply with. There may be financial requirements. The Sarbanes-Oxley compliance requirements that you have may require that you keep certain amount of data private, certain amount of data segmented off, and that you're keeping it for a certain amount of time. You also have to think about how you're going to watch this baseline over time.

When you install a new application, when you install a new patch, it may affect what you need as a minimum requirement for the security of those devices. Sometimes installing these patches creates other security holes. So you may not want to install certain patches because of that issue.

But if you don't install that patch, other things may be a problem later on. And we need to mitigate that. It's a balancing act. It's a very complex balancing act we have to think about, but it's one that we have to keep our eyes open and maintain these systems through the entire life cycle of that application and the entire life cycle of that operating system.

If you're plugged into the different security blogs, you're watching some of the announcement areas where you can gain information about vulnerabilities, then you see new vulnerabilities come out every day. For different applications, different operating

systems, there's constant, constant motion there. And so you have to keep up with what's going on. You have to also continuously monitor your systems and make sure that they are constantly up to date, that you are modifying and updating them. And you have to make sure that whatever you do on those systems, whether you are changing a patch, maybe you're adding a different configuration, that you're keeping an eye on how that changes the security posture of those systems.

The **National Institute of Standards and Technology** has created a document here in the December 2010 time frame. This was in draft form. So you may want to go out and see what the latest version of this. But this is a document for continuous monitoring for federal information systems in organizations. And although this is focused on the United States federal organizations, you may also want to look at it if you're not in a federal organization, because there may be some very good information in here on best practices on what you should do to continuously monitor your systems.

If you've created these baselines, you're constantly monitoring them and then you realize one of your systems does not match what we consider to be a baseline for the security of that system, you have to decide what process you want to take. And generally, it's something called remediation. Maybe we're taking those systems and they only have access to a very special remediation network.

So if a system plugs into the network and it doesn't have the latest antivirus signatures, we're going to make sure they can't access anything on the network. We're going to put them in a special network automatically. And that network would only have access to download the latest antivirus patches, maybe make sure they've got the latest operating system patches, but have them put in a place in the network where they cannot cause a problem.

That becomes very important if we're trying to maintain this particular security posture. And once they get the patch installed, once they get the latest version of antivirus updates, we'll, of course, constantly monitor that system. And when that computer's now up to date with the minimum security baseline, they now gain the normal access that they would have to the network.

A lot of this can be automated through 802.1x. We've talked often about network access control. And I almost always with network access control, there's a section of the network just for remediation. And that's where we'd want to have all of those security tools available so that the user finds that the disk encryption is not enabled on their computer or the antivirus is not up to date.

They'd go to one particular place on the network and still have the access to be able to fix the problem first. You want to be sure that every system on your network is running that particular baseline so that you can be absolutely sure that your security posture is the one that you want for your organization.

**Tags:** [baseline](#), [certification](#), [comptia](#), [monitor](#), [remediate](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Reporting – CompTIA Security+ SY0-401: 3.6**

If you can't see a problem, then you can't fix it. In this video, you'll learn how reporting can help you create and maintain a security computing environment.

One good way to mitigate or even prevent certain attacks from occurring is to have as much information available as possible. Unfortunately, a lot of the devices on our networks can give us a lot of metrics and a lot of details about what's going on. You've got content and information that you can gather from your firewalls, from your IPS systems, from your routers, your switches. You've got information from your servers and other devices.

The challenge, of course, is what you do with that information. If you want to be able to at least get an idea of what's going on, you'll need to decide what you want to look for. What are the metrics that are most important for you?

From a security perspective, you might want to look at throughput, you might want to see the number of authentications or attempted authentications that are occurring over a certain time frame. Maybe you'd like to watch the CPU utilization of a server so that you can see exactly when the heaviest loads might be. If a server is one that you would not expect a heavy load, that will be a good time to find out when something changed.

So we need to think about thresholds for these metrics that we've created. Do we want to see if a device is up or down? Do we want to understand what particular threshold we want to know about with that CPU utilization? Perhaps you never want to be informed about CPU utilization unless it goes above 70%, because this server should never be doing that.

Or what about temperature? What about network throughput? There are literally thousands and thousands and thousands of possible metrics that you could be gathering and creating thresholds for. But you have to figure out which ones are most important for you and your environment and the type of applications that you're using. Once you identify these metrics and you've set thresholds for them, you absolutely need to be informed when those thresholds are exceeded, and you need to find out the best way to contact you.

Maybe if you carry phone around all the time, you get an SMS message. If you're an email type person, maybe you're emailed immediately. There's pluses and minuses to any type of disposition system. So you might want to even combine different ones together so that you not only get a text page, but you also get an email sent to perhaps a number of different people.

That way, it's not just one person informed. If you exceed a threshold and it is an important threshold, you can inform a large number of people at one time. If we're busy collecting all of these data points about CPU utilization and number of authentications and identification of when the network thresholds, the network throughputs are going up and going down, it will be really great to be able to track this over time.

So a lot of the monitoring systems that you'll find also include a way to be able to create some trend reports. It's very, very difficult to get a high level, a big picture view of what's going on, unless you can put it into a form where you can look at a lot of it in one place. And these graphical representations of what's going on really do tell a story.

You can see exactly when traffic is getting higher. You can see when traffic is getting lower. You can understand why you exceeded a particular threshold. Otherwise, you'd have to pour through pages and pages and pages of log files. And at the end of the day,

you probably still wouldn't have the same perspective of things like network throughput or any of the other metrics, unless you're able to put it into a graphical form.

You want to also look at how often you're going to monitor these devices, what timeframe you want to report on this information. Maybe you want to poll a device every minute to get a metric from it, maybe every five minutes, maybe every hour. And of course, there are advantages and disadvantages to doing either one of those things. You also need to think about what type of reports you want.

Do you want to daily report that gives you a representation of what happened the previous calendar day? Or would you like a roll up at the end of the week that shows for the same seven day period, tell me an idea of what happened during that seven days? Obviously, the more data we put into a single report, the harder it is to get a lot of the granular your out of it.

But now you can make a decision about just how much data is important for you and exactly what type of information you would like to be able to see over what time frame. Just remember that when you start collecting data, you are absolutely going to be collecting a lot of information. So very often, these visualizations tools, these polling devices that we have are able to age out the information as we go.

So we might keep one minute intervals for 30 days. But after 30 days, take this one minute intervals and average them out to an hour and simply store the hours information. That way, we're getting rid of a lot of data points and a lot of storage and really summarising information over a longer time frame. Sometimes you'll need to be able to set those particular roll ups yourself, those aging out of that data yourself. And you need to think about how long you need those raw statistics.

If somebody six months from now wants to know a minute by minute breakdown of what occurred during that time frame, then you're going to need to keep that raw data over a much longer period of time. And think about also exactly what security metrics you would like to look for. You want to be able to understand if there was an increase in malware activity, if you're getting more spam coming into your environment, maybe your spam reporting system can tell you that there's a big uptick in spam.

And maybe that will put you on alert for a little more phishing activity. Maybe you want to see how many devices on your network have received the patches, or perhaps more importantly, what devices did not get patched in the latest update.

And sometimes, an increase in bandwidth can lead you into more information about what might be going on in your environment. A good example of this is in May of 2011, a company called **LastPass**, they create a digital wallet where you can store all of your passwords encrypted in one place. They were looking at their logs, they were looking at their reporting system, and they noticed an anomaly in traffic that was increased from a particular server.

This is a server that contained sensitive information. It contains our password data. And they noticed that there was a little bit of an uptick in the amount of traffic transferred. And of course, when they see that, they take into account perhaps the internal systems that they have. There's backup systems and testing systems that they use.

And they went back and looked at their logs and realized this wasn't us. This was not our internal systems. This perhaps could have been someone else in our systems transferring information, sensitive customer information. And that was a bit of a problem for them. So they were very forthcoming.

They got a public message out on their blog. They tweeted information about this with all of the details of what they saw. They had no evidence that there was a bad guy who'd gotten this information. They had not seen this information being used elsewhere.

The information itself was encrypted. But it pointed to the potential for problems. And they wanted to mitigate any issues. So they required immediately that everybody change their password. If you use the **LastPass** system, you're going to need to change your password immediately so that if somebody could grab that encrypted data and then decrypt it and find all of our passwords, they could try them. But they still wouldn't be able to get in and use the information stored on LastPass.

But that certainly spoke to a much bigger issue. And when you're thinking about monitoring your systems, finding out different thresholds and understanding what to look at over time, think about those security concerns and what you can do to help prevent some of these security problems.

**Tags:** [alarm](#), [alert](#), [certification](#), [comptia](#), [reports](#), [security](#), [trend](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Detection vs. Prevention – CompTIA Security+ SY0-401: 3.6**

Should you detect, or should you prevent? In this video, you'll learn the advantages and disadvantages for implementing these security techniques.

When we think about protecting our assets, we're often in a position where we need to consider adding a technology solution and a non-technology-based solution. And a good example of detection versus prevention comes in to play when you're trying to protect a certain area of your organization. A good example is when you're trying to decide, do I put a camera here, or do I put a physical guard a physical person in place?

Obviously, a camera is very easy to install. It's relatively inexpensive to do this. But it is very difficult to proactively prevent somebody from doing something. A camera isn't going to stop someone from walking down the hallway. And occasionally, you don't even recognize the camera's there. You don't even think that there may be a camera in place watching what we're doing. So it's not a very good deterrent of keeping people out of a certain area. It is extremely valuable, of course, after the event. You'll be able to see exactly what happened in a room, in a hallway. All of that is saved for later.

If you have a guard in place, that's a completely different posture. It is now a physical person who can interact with the people walking down a hallway, going into a room. And it allows you to proactively prevent certain things from occurring. And obviously, there are disadvantages to this as well. It's a lot of cost to have a person in place. You have to pay a salary. You have to make sure that person is there. You have to staff, perhaps 24 hours a day, seven days a week. But it does have a personal impact. And when you're trying to decide, do I allow people through and simply detect that they're there, or do I actively prevent people from doing something? We now have a better idea of which solution to have there.

In many cases we may have both. We may need an immediate, proactive, stop people from going in a room. But we might also want a way to go back over time to last week, to last month, to last year, and see who went in and out of that room. And who did we sign in, who did we sign out, to back up perhaps what a guard was there being able to do themselves. You've now got decisions to make, not from just a person perspective, but there are also a number of technology solutions that we should consider also, when it comes to detecting versus preventing.

A very common technology solution when you're trying to detect the bad guys coming in to your network, is an intrusion detection system. We've talked about these before. You have people out on the internet that are using resources in your environment. You've also got your end users in your environment that are accessing resources on the internet. And the traffic is flowing back and forth through your routers and your switches in your

environment. And a lot of information is passing. We have multi-megabit, in some cases gigabit connections, out to the internet.

There certainly is a lot of information flowing through the very fast networks we have today on the inside of our network. Intrusion detection systems were built to be able to watch all of that. These are usually devices that are put off to the side. Information is copied off to the intrusion detection system. Usually it's done through a built-in system, inside of the switch, that allows us to send a copy off to a separate port. Sometimes it's a physical tap, where you are connecting right in to the middle of a connection and simply sending a copy of that data to the intrusion detection system.

One of the challenges, of course, though, with intrusion detection is that is only detecting and alerting on these things. There are some intrusion detection systems that have some limited capabilities in actually stopping that traffic. But because we're getting a copy, or a mirror of the traffic, that is a very anemic way to try to control information. It really is just there to alert you.

And the idea is the intrusion detection system would see someone communicating to a server. And someone from the internet is trying to take advantage of a known vulnerability on that server. Maybe it's something that is not patched through the Microsoft operating system. Maybe it's a vulnerability in the database, someone trying to do database injection. The intrusion detection system would see that. Because it's seeing all of the traffic going back and forth. And then the intrusion detection system can then inform you and let you know that at this date and time, I logged in event that occurred, that this particular IP address from the internet was accessing this particular web server, and they perform this particular vulnerability attack. They went after the database injection. They went after a known vulnerability in that Microsoft operating system. And having that in place can at least give you a list of what has occurred over time back and forth over your network.

These days, our threats are happening much more rapidly. We have a lot of traffic going back and forth. And we want to be able to stop these problems as they are occurring. Instead of having an intrusion detection system, we have mostly moved these days to intrusion prevention systems. These prevention systems go directly in-line. So all of the traffic flowing in and out of the network, or even inside of the network, has to go through and **IPS**.

The **IPS** is now responsible for identifying any of these problems when they might occur. If someone from the internet is trying to take advantage of a known Microsoft operating system vulnerability, the IPS will see it, and because it's in-line, it will stop it. So even though the bad guy sends the traffic through, the IPS identifies that traffic and right then, drops the packets. They never continue through the network. Someone can try to do a database injection. The IPS recognizes database injections, is able to stop it right there.

Obviously being in-line is a very critical piece. There's usually redundancy and fault tolerance associated with the IPS implementation. And IPS systems must be able to keep up with the speeds and still understand all of the different vulnerabilities and signatures that it needs to be able to stop the traffic.

But now you've got options. You have the ability to simply report on what's going on with intrusion detection. Or you've got the ability to go actively in-line and stop these vulnerabilities from going through our network with intrusion prevention systems.

**Tags:** certification, comptia, detection, prevention, security

**Category:** CompTIA Security+ SY0-401

### **Vulnerability Scanning Overview – CompTIA Security+ SY0-401: 3.7**

How many vulnerabilities do your network devices have? In this video, you'll learn how to use a vulnerability scanner to find the susceptible areas in your network.

If you're looking to find the vulnerabilities that might be on your network, a vulnerability scanner is a great way to go about automating that process. One of the challenges we have, of course, is that there are vulnerabilities identified practically every day, very often multiple vulnerabilities a day. In the United States, there is a national vulnerability database.

If you go to [nvd.nist.gov](https://nvd.nist.gov), you'll be able to get an idea of some of the vulnerabilities that people are identifying. If you go through this list, you'll notice there's vulnerabilities for operating systems, there's vulnerabilities for applications. Certain services have vulnerabilities associated with them. And we're discovering new ones all the time.

So think about all of the systems you might have in an enterprise environment. There are many different operating systems. There are many different applications. It becomes now very complex to be able to keep up with all of this and understand what's going on. Fortunately, these vulnerability scanners are designed to keep up with all of these latest vulnerabilities.

They're able to understand how susceptible your systems might be to some of these vulnerabilities. And they can really go through and query a system, in some cases even try the vulnerability itself to see if it can take advantage of a system. So just turning on a vulnerability scanner—probably not the smartest thing to do.

I've been in environments where somebody turned on a vulnerability scanner not recognizing all of the different and varied systems and ended up bringing a number of very critical systems down. So be careful when you start going through and proactively scanning all these devices. You don't want to create more problems than what you're trying to solve.

Not all scanners are alike. Certain scanners are very good for general use. Some are focused on applications. Others are focused on certain operating systems. You'll need to look at all of the different scanners that are available to you and see which one fits best with what you're trying to do in your environment.

A good example of some very common scanners are things like Nessus, Nikto, Nmap. You've got SATAN, and SAINT, and SARA, which are very similar. They were built from similar systems there. And it really just depends on the need you have to be able to scan and identify these vulnerabilities in your environment.

I'm running a vulnerability scanner on my network. I'm running the home version of Nessus, which gives me the ability to run and look at all the different systems that are on my local network. I have quite a few systems on my local network. You can see them listed here. It has identified the total number of vulnerabilities that it has identified in these systems.

Some of them are very high vulnerabilities. Some of them are medium category. And others have a severity level of low. And you can also see how many ports are open on those devices. So not only has this device identified vulnerabilities that happen to be known, it's also identified opportunities for people to be able to connect to these devices.

Let's look at one of my devices. This is the device I'm running right here. And it's showing me that I have indeed some vulnerabilities— a high level critical vulnerability that shows a vulnerability in Microsoft Office. So just by running this scanner on my network, it went through and found every system that was on my network. It automatically went through and ran a series of tests on all of these systems and created a report that I can now

reference to understand what are some of the vulnerabilities I should be aware of in my environment.

One thing you do have to watch out for with these vulnerability scanners is the scanners aren't perfect. They don't really have context as to the types of systems that you have. They really start with a pretty blank slate and they build as much information as possible there. But they can be a little bit fickle. They can not quite exactly hone in on specific problems.

So you do have to go back over the results and make sure that the vulnerability scanner really is giving you correct information. For instance, if you are in an environment where you have network level devices— you might have packet filters in place, you might have firewalls in place— you may not have the ability to go from one side of your network to the other without having a system in between to watch what's going on.

So of course, your vulnerability scanner will be affected by this. If you can't reach a device or you're filtered from going to that device, then obviously I can't check it for certain types of vulnerabilities. So that's something to keep in mind. You also have the devices themselves that might have their own personal firewalls.

There might be different application versions running on systems. Maybe the operating system itself doesn't lend itself to being able to do a very good vulnerability scan. And sometimes you can adjust your vulnerability scanner with specific logins that might give you extra access to devices. And if that's not enabled and turned on, it may not be able to get to a device and really do a thorough vulnerability scan of it.

So make sure that you look through the results. And occasionally, you will find some very surprising results, things you didn't know about your systems out there in the field, things that you thought were enabled it turns out were not enabled. Security systems you thought were in place perhaps were not in place. So make sure you go through your results and be able to identify when some of these surprising results might be.

That way, you can go and resolve those issues on those systems or on those networks. One of the surprising things on my network was a Windows system that I was not expecting to see with a vulnerability of high listed here— my 192.168.1.19. If I drill down into that and look at the vulnerabilities associated with it, it says this is a Microsoft Windows SMB shares unprivileged access.

And it's certainly set to a severity of high. That doesn't sound very good. If I drill down into it further, it says it is possible to access this network share. And it says it's able to do it without any specific kinds of rights. And in fact, it said it was able to get in and read all of this information from the hard drive.

The entire C drive is readable and writable across the network. And it's probably a system I had my lab. I configured it a certain way and completely forgot that I'd set it up for such an open type of access. But by running this vulnerability scanner, it didn't find a known vulnerability. It found a known misconfiguration on that device. And it informed me that if you'd like to make this more secure, you might want to think of setting some permissions on this.

That's the value we have by running these automated vulnerability scanners on my network— being able to have it go through and find every system for a myriad of different problems, vulnerabilities, and configuration issues can really help secure your network even further.

**Tags:** certification, comptia, scanning, security, vulnerability

**Category:** CompTIA Security+ SY0-401

### Assessment Tools – CompTIA Security+ SY0-401: 3.7

As a security professional, you need assessment tools to help keep your network secure. In this video, you'll learn about active vs. passive tools, protocol analyzers, honeypots, and more.

What devices are connected to your network, and what operating systems are running on those devices? And of those operating systems, do any of them have any security vulnerabilities that we need to know about? Well, one way to go about answering these questions is to use something called an assessment tool on your network.

One type of assessment tool is something that can gather information passively. These would be tools that don't interactively log into devices. And they're not trying to break into a device using a vulnerability. Instead, these passive devices try to gather as much information from the outside without directly interacting with those devices.

Something like a packet capture is a good example of a passive assessment tool. But if you really want to go after a system and really see how much information you can get by knocking on the door or trying to see if all of the windows are open, you can use something like an active assessment tool. These devices are things like vulnerability scanners that are configured to log in to devices to see what might be inside of that machine.

They could be things like honey pots or port scanners or even devices that are designed to grab banner information when you first connect to a device. These are actively logging in and actively interacting with those devices. So we put them in the category of an active vulnerability assessment tool. Protocol analyzers are certainly a valuable tool to use when assessing what's happening on the network.

They're very passive. Since they're watching all of the packets go back and forth, we sometimes will see these referred to as sniffers. But of course, the term sniffer is a trademark name from **NetScout Systems**, but we still generically call it a sniffer in many cases. This is really gathering everything from the network.

So all of the traffic that goes by is gathered, captured into memory or on to disk, and we're able to go back and see what happened on the network when all of that traffic was going back and forth. One very popular open source version of a protocol analyzer is Wireshark. You can download this for free, load it on your system, and begin gathering packets immediately.

If you really want to see the way an application interacts with your computer or you just want to have an idea of what's going on across the wire, then Wireshark is an excellent tool to use. If you're interested in knowing the way an application works across the network or you're just curious about traffic that may be going across the wire, then a protocol analyzer like Wireshark would be an excellent tool to use.

And it's remarkable how much information is going across the network that is completely in the clear. You can gather information about where people are going, what they're surfing the websites they connect to, email information, and even passwords can be found just by analyzing the traffic that goes over your network. Vulnerability scanners are a very useful tool to try to see if there are any problems with applications or servers or operating systems that you might have in your environment.

This is going to give you an idea of where problems might be. Application vulnerability scanners are focused in the way that applications operate. So if you're trying to find a cross site scripting problem or you want to check to make sure there's not a database

injection vulnerability, then an application scanner would be an excellent choice. Operating system scanners look at the entire operating system, not just the applications that are going over them.

And if you're thinking about those monthly updates you get from Microsoft that are telling us to patch our systems, and every month we get a series of patches from Microsoft, it's these types of scanners that are going to be able to tell you if you are completely patched up or if there are any holes that a bad guy might use to take advantage of one of these known vulnerabilities. There are a lot of different options for both application scanners and operating system scanners.

Certainly commercial scanners are available. There's also a number of open source scanners. It just depends on what you would like to be able to scan and just how much information you would like to gather from these scanners. There's a number of different vulnerability scanners that you can download and try for yourself.

One of the scanners that's been around for a very long time is called **SAINT**, used to be called **SATAN**. And it's one that you can download and install and run on your system. One that is licensed for home use to use absolutely free is Nessus. And Nikto is a very good application vulnerability scanner. All of these have advantages and disadvantages.

But if you start running one of these scanners, you'll start to understand exactly how much information you're able to see by simply scanning all the devices on your network. Just like bears are attracted to honey, the bad guys are attracted to honey pots. These are systems that we would install into our network. They look like an absolutely legitimate machine, a server that might be running in our environment.

And it might even have a door that we've simply propped open a little bit just so the bad guys can get in to see what's going on. And the idea is to get them inside of this system and trap them into what might be happening. A single standalone device we call a honey pot. So the bad guy connects to this single system. And now he's looking at file information.

He's going through the file system. He's trying to log on or even take care of a vulnerability, not understanding, of course, that this system is one that we completely created just so we can trap him here while he performs all of these vulnerability checks. If you want to get a lot of honey pots together on your network, you would have a honey net. And now you can bring the bad guys in and get them moving back and forth between many different systems all at the same time.

If you'd like to install a honey pot on your network, or see what other people are doing, you can go to the Project HoneyPot website at [projecthoneypot.org](http://projecthoneypot.org). Port scanners are used to try to determine what type of open ports might be available on a system. So if you want to see what a firewall may be passing from a port number perspective, you would want to use something like a port scanner.

This is also a good tool to use if you'd like to identify what an operating system might be or what a specific application version might be running on that operating system based on some of these open port numbers. You can very often determine all of this information without ever logging in, authenticating, or running that particular application. If you've ever used a port scanner before, you know that if you're doing a TCP port scan, that it's using this three way handshake to be able to see if those ports are open.

Now, take that same idea, do it across many thousands of port numbers on a device. And then do it across all of the devices on your network. And you'll see how a port scanner can be a valuable network reconnaissance tool. Another useful reconnaissance tool that can be used for assessment is one that can grab banners from the services that might be running on a machine. You've probably seen these before if you've ever SSHed into a server or you connected to it with a web browser and you looked at the header.

You'll see there's a lot of information that is sent back to your machine, even though you've not even authenticated to that application. The banner is always going to be there, because the application is configured to always provide that banner information down to the client. Sometimes it's behind the scenes. It might be in the header of HTTP information.

But if you grab the information from a protocol analyzer or you view a specialized tool that's designed to grab these banners, you'll be able to see all of this information. On the screen, I grabbed a banner, really an HTTP header of communication that goes back and forth when you connect to professors-messers.com from your browser.

And you can see a lot of information inside of this. One at the very top, though, is the type of web server that you are connecting to. Now, I have a reverse proxy that you connect to before you ever hit the Professor Messer website. And that reverse proxy is running at a place called Cloudflare. And you can see the engine that's running here is engine x.

That is the web server on that reverse proxy. These are the types of details that the bad guys will gather from all of your systems to try to determine if there are some known vulnerabilities they can go after on these devices.

**Tags:** active, banner, certification, compTIA, honeynet, honeypot, passive, port scanner, protocol analyzer, security, vulnerability scanner

**Category:** CompTIA Security+ SY0-401

### **Assessment Types – CompTIA Security+ SY0-401: 3.7**

It's important to accurately categorize security assessments. In this video, you'll learn the differences between a security risk, security vulnerability and security threat.

If the assets in your organization can be compromised, it's probably through something like a security risk. This would be an event that causes those assets to be at risk. This is something that can be a active event where somebody's trying to break into your organization, or it can simply be a circumstance, perhaps an act of nature that causes a fire or a flood. To be able to properly guard against these security risks, then we need to understand what they are.

We need to create a list of those and find out what would be our first steps at guarding against those risks. If we're worried about people breaking in stealing things, then maybe we should think about how we prevent that physically. If we're worried about having an act of nature come through and cause a fire or a flood, then we need to put things in place that would help reduce that security risk.

From a physical perspective then, maybe we want to be sure that everything is behind a locked door. Maybe we want to be sure those door locks are connected to our badging system. Perhaps we put a guard posted in front of those physical doors, and we make sure that all our visitors have badges and they have to also use those badges to go in and out of that particular locked area.

These processes and procedures can also be technical. We want to be sure on our internet connections that we have a firewall that will protect all of the assets inside of our organization. And on every single computer perhaps, we'd also like to have antivirus or anti-malware software running so that the bad guys can't put a piece of malware on our machine and begin extracting the files off of our computers.

The security risk itself might be a vulnerability. If we've managed to put a door in place but we've not properly locked that door or maybe we have disabled some firewall rules on our internet connection, then we've created a problem that is obviously going to be a

big vulnerability in these systems. This is something that you might find out about every month.

Microsoft, for instance, releases an entire set of security patches every month for all of their operating systems. And so they're advertising to the world, here is every place that there might be a vulnerability inside of our operating systems. And so we want to be very quick about patching those operating systems to prevent the bad guys from taking advantage of those vulnerabilities.

Sometimes the vulnerabilities themselves will never be discovered. You might be running an operating system right now that has a vulnerability that nobody knows about. So obviously, there's not going to be an announcement about it until somebody discovers that vulnerability. The vulnerability itself, of course, isn't a problem. If nobody ever discovers the vulnerability, obviously nobody knows to take advantage of it.

If you walk by and the door is closed and you never try the door to see if it's unlocked, then you'll never know that the lock is broken and nobody's ever going to go into the room. But of course, we're always concerned when a vulnerability exists, whether it's known or unknown, and we have to plan on protecting our systems regardless of the situation.

The threat is the thing that we're most concerned about, because that's what's going to take advantage of one of these vulnerabilities. If somebody's walking around and trying every single door, they will eventually find the one that does not have a working lock. The threat may not be intentional. It may be something accidental like a fire or a flood.

So we have to plan for every contingency. We often call the person who's trying those doorknobs or trying to break into a computer system the threat agent. They're trying to take advantage of one of those vulnerabilities. And they do this by using a threat action. They're either trying to perform a buffer overflow to that operating system or they're simply trying doorknobs.

But those are actions that will ultimately take advantage and exploit those vulnerabilities. The result, of course, is a loss of security. And now someone will have access to a room that was unlocked, or they'll have access to all of the files on your operating system.

As a security professional, it's these threats there we're always mindful of. So it's always important to understand what the risks are in your environment, know where the vulnerabilities might be, and by doing that, you can help prevent some of these threats from ever occurring.

**Tags:** [certification](#), [comptia](#), [risk](#), [security](#), [threat](#), [vulnerability](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Assessment Techniques – CompTIA Security+ SY0-401: 3.7**

Security policies should be written to include the security assessment of your infrastructure. In this video, you'll learn the best practices around baselining, reviewing code, performing design reviews, and completing architecture reviews.

An important technique for assessing the security in your environment is baseline reporting. You need to know where you are so that if you do have a breach or you're considering adding on additional things to your network, you'll have an idea of where you are today. So you want to determine what your risk is right now and be able to compare it should there be an event, should there be a new installation of an application, should there be new installation of patches, anything that we change so that we can compare it before and after.

We also need to think about what metrics and what type of statistics will be nice to monitor. It would be good if we got some feedback day to day on where security concerns might be. But the only way to do that is if we know what the security is today. How many people authenticate to the network at any particular time? How often do we see people authenticating and not putting in the right username and password?

We have to know what that is today so that if we see a large number of those, we'll be able to determine, oh, that's completely normal, or we might be able to determine that's very very different. And then we can start researching further. You have to consider also this baseline may be appropriate today, but that baseline may change in the future. And of course, things happen in the normal course of doing business.

You may have a month end process that occurs. You may have a quarter end process or a year end process that occurs. And of course, equipment is moved in and out all the time. New applications are loaded all the time. You have new hardware that comes into the environment with new operating systems.

All of these things have to be considered. So you may end up having to create new baselines all the time. Another important security technique is to review the code that you're using in your organization. This obviously is code that you would have access to. Not all applications will give you their source code.

You have to rely on the manufacturer of that application to be able to keep up with any security concerns and to make sure that their code is something that is secure. But for the code that you're able to see internally, it's things that you've written yourself, perhaps source code that you've downloaded from the internet and you're using yourself, you're able to go through it and see what is in that code specifically.

So you may need some development experts or people that are really familiar at understanding exactly this code. And they're going to go through the code in a lot of detail. They'll break it up into separate functions, into separate sections, and be able to go through it with a fine tooth comb and really give it a detailed overview of what's in there. What they're looking for is any part of the code that would open you up for a security vulnerability.

So you want to make sure that if there is any input into an application that you're validating that input, that nobody can do any cross site scripting, that nobody can do any database injections. You want to be sure that any other type of input or output coming from this application is examined. And that's where you really need to get somebody very familiar with the security concerns that we have today that also understand the application software and programming language that you're using.

That's a great combination, because they'll be able to go through the code, validate everything that you have in there, and in certain situations, make sure that that code is not going to be susceptible to those types of security concerns. Usually when people are assessing the code, they also need to understand how your software work.

So there may be a design review of the software itself so that people understand exactly what this application is supposed to do. Everybody understands what the input might be for certain fields. They understand what the output is expected from this application. So you really have to go through how an end user is using this and how we've designed the software to work for that end user.

So many ways can a user input data? Does it have a lot of different fields that somebody might add information into? Every single one of those fields may need to be validated for the input that's put in there. You have to think about the way that the bad guys would use this application against you and think about every possible attack vector, every single place a bad guy might use to get into your data.

All of these attack surfaces, whether they're something it's obvious, like a field, or something that's behind the scenes like an API, needs to be considered. So part of your design review is really going to be looking at the details of this application. That's why lot of people don't want to add APIs or they don't want to allow people API access, because that opens you up for security concerns there.

That's why we want to keep that attack surface very small. If we're able to limit what people are able to do in the application and the type of input that's there, then we're also limiting what the bad guys can do with that application. So your design review should look at the balance between the usability of that application and the security concerns that are associated with that usability.

Of course, an application isn't just about the code itself. We have all these other systems that revolve around the application. There are web servers, there's database servers, there's application servers. There maybe middleware that talks to a mainframe. There's so many different components to the way these applications work. So we also have to understand every step along the way.

We have to think about the security concerns for each one of these devices, for the database, for the web server, for our clients themselves in the application that's used by our clients. If you look at the application and what we're thinking about from a security perspective, it's those very broad security requirements that we've talked about before. We need to think about confidentiality, we need to think about integrity, and we need to think about availability.

And we apply each one of those to every single component of the application. SQL is obviously a standard, but just because there is a standard in the way that we talked to the database engines themselves doesn't mean that there aren't different security concerns between the different database engines. A MySQL database interacts with the operating system and has different security concerns than for instance of Microsoft SQL Server.

Both of those use the same SQL language, so we have to not only think about the language that's used into those databases, but do those databases have very specific vulnerabilities and things we need to think about from a security perspective. We also think about the end user as using a browser. But browsers are very different as well.

Internet Explorer is different than Firefox, is different than Safari. Each one of those browsers has different security concerns as well, and different vulnerabilities that people have found in those browsers. So don't just look at your code for your applications. Think about all of the different components of your application and how they all work together so that you can get a very good idea of the security related to the entire architecture.

### **Penetration Testing – CompTIA Security+ SY0-401: 3.8**

One way to determine the security of your network is to actively attack your computing resources. In this video, you'll learn techniques for testing security controls on your network.

If we're performing a penetration test on our network, then we are going to be actively attacking the systems that are out there. You'll sometimes hear this referred to as pen testing. And this is a little bit different than vulnerability scanning. When we were doing vulnerability scanning, we were being relatively passive. We weren't really attacking systems directly.

If we're trying to get into a system, though, and really see if we can take advantage of the vulnerabilities of that system, then we call that penetration testing. If we're able to get in and see what the bad guys can do, then certainly we can test and be sure that we're putting the right security in place. There's usually a mandate also in some environments that someone come in every so often and actively attack your systems, actively try to get in to your data, to your operating systems, and in through your applications.

There are some guides out there that can help you, one from the National Institute of Standards and Technology.gov. Here's the publications **URL**. This is a technical guide to information security testing and assessment. So that is a nice read that can help you understand some of the techniques and some of the things that you can use when you plan to do some of your penetration testing.

If you're going to be doing penetration testing, then you need to understand what the latest vulnerabilities might be for those operating systems. So you have to stay up to date with what the latest threats are. The challenge, of course, is the applications may have been around for a long time. But sometimes we're just finding new threats in some very old applications.

And if you don't keep up with that, then you may never know that there was a new way to get into the way that that application works. You can look at a big list of these. The **National Institute of Standards and Technology** has a **National Vulnerability Database** that you can find at [nvd.nist.gov](https://nvd.nist.gov). Another thing you can do is to constantly do vulnerability scans.

Use those new methods that you've now learned about to see if any of the systems on your network are susceptible to those new threats. And you want to make sure that whatever scanner that you're using is using the latest signatures because of that. That way, you can say up to date with all of those and watch the news.

There is all kinds of interesting information occurring for the latest set of news. If there's something big in the news, the bad guys are going to try to find ways to get in using some of those new techniques. And if you're able to see new vulnerabilities come out, then you may be able to set up your firewalls. You may be able to set up your systems so that even if somebody tried to get in with some of those new techniques, they would not be able to.

There are many aspects to penetration testing. One of the things that we can do is to bypass some of the security controls that might be in our environment. Go outside your environment. Try to force your way in through your firewall. Can you get into the network from somewhere out on the internet?

You might also want to think doing this in person. Think about going and trying to get into the building to get around some of the security you might have. Are there certain gates that are left open? Are certain doors easier to get into? Perhaps there's an area of the building that's not monitored.

You want to be able to find those things. You have to think like the bad guy to be able to do that. There are also people inside the organization that may bypass security controls. So don't focus solely on getting in from the outside. Think about how the people internally in your building and in your network have access to these different systems.

Do they get around your database controls to grab information? Are they taking information from those databases and sharing them with third parties using different tools, like sending it in a Google Mail or sending it through a Yahoo Mail? Your penetration testing should consider this lack of control and see just how much you're able to do if you had a simple login like everyone uses on your network.

You also want to think about using the same tools that other people use to get around your security systems. There are things like Ultra Surf and Tor and many of the other proxies and encrypted methods to get around your existing security controls. If you have set up a policy that says you may not send our sensitive data out through Google Mail and you're blocked Google Mail, people try to find a way around that.

So use some of those proxies, use some of those encryption techniques. See what you're able to boot from inside of your networking and you'll have a pretty good idea of what anybody else is able to do as well. When you're testing your security controls, try to use the same methods that the bad guys use. Try to get in your firewall, get around your IPS system, maybe try to do some scanning.

How slow do you have to scan to get through your IPS? That's one of the best ways you can get an idea of what the bad guys would have to do to get into your systems. And try many different techniques. Maybe it's not just one scanner. Maybe it's other scanners. Maybe it's other security frameworks that would allow you to try different methods to get in.

This is going to give you a pretty good idea of what the bad guys are seeing so that you can have an understanding of how you can set up your security systems to prevent them from even getting in in the first place. When you're at the point where you're really trying to exploit some of these vulnerabilities, you have to be pretty careful. Some of these buffer overflows or injections can cause the application to break.

You may cause the database server to be unavailable. You may cause the application itself to not be accessible through a web browser. So that denial of service, that loss of data can be pretty bad. So usually this is something you're planning internally and you've got backups, you make sure that everybody has systems in place that are aware that this is going to occur.

You want to make sure that you don't cause a problem for anybody else, especially with your production systems. And you may need to try different methods to break into a system. Try your brute force attacks. Try your buffer overflows. Try the known injection types for that application.

There's many, many different ways. And your goal, of course, is to see if you can get in and pwn that system. And if you're able to get in, then the bad guys can get in as well. And that's what you're trying to avoid. So if you can get in, then you know exactly what you should be patching and filling in all the holes that the bad guys could possibly use.

When performing a penetration test, there's this concept of a black box, a white box, and a gray box. And it refers to how much you know about this network that you're attacking, how much you know about the databases and the systems and the firewalls and all of those things that might be in place. If this is a black box test, then you approach it from the perspective of knowing nothing about what's behind your IP address or what's on your network, as if you showed up with no prior knowledge of anything that's happening inside.

Sometimes if you have a third party that you're contracting to do a test, this might be a way that you start to say, try just getting in and seeing what you can gather. Try some recon. Try to figure out what systems are there. And then try to attack them knowing nothing about what might be there already.

The exact opposite of this, of course, would be a white box, where you're giving someone a network map. Maybe you already know the IP addresses of your database servers. You know what version numbers they happen to be. And you might be doing some very, very specific vulnerability checks, some very specific penetration tests against all of those different systems that you might have.

A gray box obviously then would be something in between. You know a little bit about the network, you know a little bit about the systems, but you don't know everything. And if somebody's going to start performing some tests, they may be much broader tests to be able to determine exactly what might affect those systems that you have in your environment.

A common example of a penetration test is somebody using some specific well known vulnerabilities to attack the operating systems that you might have in your environment. I have here two virtual machines that I'm running. One is my canary machine and one is my destiny machine. This is my machine that's going to be doing the attack.

The Windows machine here is a relatively unpatched version of Windows. Let's see the IP address on this machine. It's 192.168.1.16. And it's just sitting out there on the network somewhere, waiting for somebody to try performing some type of attack. And see if we can penetrate the security that's on this Windows machine.

Now, I don't have the login into this machine as the bad guy. But what I'm going to do is run this **Metasploit** framework that I'm running here. And I'm going to choose a very specific Windows exploit that deals with the **RPC**, the **Remote Procedure Calls**, in this Windows. And I happen to know there was a very, very bad, very, very common vulnerability in some older Windows systems that took advantage of this.

This particular Microsoft **RPC DCOM** interface buffer overflow is one that I can use to try to attack this machine. And you can see there's a lot of different things we can do here. It's definitely going to this system. And I have the option now of setting up and running this particular exploit. So let's choose some of the things associated with this exploit and run this injection.

I'm going to run— let's do a reverse shell here. I'm going to do a shell reverse **TCP**. And 192.168.1.16 is the one— the 25 is my list and the remote host is 26. Let's scroll down and put that in. So we're going to add the IP address of that Windows machine— 26.

And I'll just choose to run that exploit in my console. So it's going to try going out to that Windows machine, attacking it with that specific vulnerability. And now in my console, I have a shell. And it's one that allows me to interact with that Windows machine. I can do a directory. I can change directories off to the root.

And now I'm in the root. Let's do a directory of that. I'm on that Windows machine. I have access to everything on that Windows machine because of that vulnerability. And I would have only been able to see if that machine was really, really exploitable if I tested it with my penetration testing. Now, the idea is I would get the patch for that Windows machine. I would apply that patch.

And then I'd run the same penetration test again. And we want to be sure that none of the bad guys can do exactly what I did to gain access to any of our systems.

**Tags:** blackbox, certification, comptia, greybox, penetration  
testing, pentest, security, white box

### **Vulnerability Scanning – CompTIA Security+ SY0-401: 3.8**

Vulnerability scans can provide you with a wealth of information about your network security. In this video, you'll learn about different scan types, how to identify vulnerability, and how to interpret scan results.

**Vulnerability scanning** is something that is generally a passive test. We're not connecting to a device and trying to log into that device or take advantage of a vulnerability that might be on that device. Instead, we're doing everything from the outside. A good example of this might be something like a port scan, where we're not logging into the device, we're simply sending one message to the device to see if we can get one single response back.

There's no authentication. We're not using any particular application. We're just looking to see what's accessible on that machine. This, of course, will help us understand what devices might be on our network. And if we're communicating across a distance, we may be able to tell if there's any security devices between us and the destination station. This is often a test that we think about running from the outside so we can tell what ports might be shown to the public world over the internet.

But if you run these tests on the inside, you can also get a very interesting perspective of what devices are on your network and how open they might be to everybody who's on the inside of your network. These vulnerability scans allow us to gather a lot of information. And as you're running it, there will be a lot of details in the logs.

The important part is to store as much as you can. And later on, you'll be able to go through all of the information to try to understand exactly what you saw during the scan. The scanning software and hardware that we use on today's network is extremely powerful. It uses a lot of different techniques to be able to see what's happening on a system. Generally, with a vulnerability scan, we're performing non-intrusive scans.

We're simply gathering information. We aren't actively trying to log in or exploit a vulnerability. There are also scanners that can perform intrusive scans, where you ask it to log into an operating system by giving it a username and password just to see what it's able to do. Or you tell it to try to take advantage of a known vulnerability to see if that particular device might be susceptible.

Usually, these scanners can be configured not to use any type of credential. Just assume we are a stranger from the outside with no special access to a system, just to see how far we can get with these types of scans. The credentialed scans might give you a little more detail, because you're actively logging onto a computer. And then from the inside of that computer, you're examining for instance, how many patches have been installed to that particular operating system.

It's all of these different options that give you a lot of control when you begin doing vulnerability scans. Let's run a vulnerability scan on my network and see what we can find. I'm using a product called Nessus Home, which is a free tool that you can use for doing home type scans. I'm going to perform a new scan. And I'm just going to tell it this is a test scan.

No description, and the targets on my network I'm going to tell it to use the entire range of the network that I have running, which is 10.1.10.1 through 10.1.10.254. That looks good. Let's launch it.

And at this point, behind the scenes, the scan is taking place. We can drill down on it to see what is it able to find during the scanning session. What information and devices are now appearing on my network. And then are there any vulnerabilities associated with the devices on my network?

And you can see it's actively going through the network, actively scanning for different port numbers, and identifying not just the IP addresses of those devices, but just how many vulnerabilities may be existing on those devices themselves. Now that this scan has been running for a while, we can see all of the different IP addresses that are on my internal network.

And most of the vulnerabilities that have been identified are these blue informational vulnerabilities. But you can see a number of devices have low, medium, or even high vulnerabilities associated with them. If we drill down into a device, we can, for instance, see a number of these that have been identified. And we can drill down on those. Let's do one that says, the SSL certificate cannot be trusted.

It explains what this vulnerability happens to be and how we should be concerned about how the configuration of this device is set up, especially as it relates to the SSL certificates on this device. So by simply clicking a button and giving an IP address range, we're able to gather a lot of information on where possible problems might be with the security of our network.

As you can see, the scanner is looking for a lot of information. But it can only find the things that it knows about. And a scanner generally has a database of signatures that it knows to look for in these different devices and operating systems. Generally, these scanners will have an update process so that you can have the latest signatures in your vulnerability scanner.

Almost all of these vulnerabilities can be listed and categorized online. The National Institute of Standards and Technology has a great database at [nvd.nist.gov](https://nvd.nist.gov). And if you go to Microsoft's website, they always keep a list of all of the Microsoft Security bulletins, along with a lot of technical details that can give you some history in some inside into the severities of these vulnerabilities.

Sometimes the scanner will give you a very generic response, saying that there may be a particular kind of vulnerability. So it's still up to you to do the final checks to make sure that what the scanner is telling you is really accurate for that computer. The scanner is at least going to give you a heads up and let you know that a problem may exist.

But ultimately, it's up to you to really make the final determination. The results of the vulnerability scan can give you a lot of work to do. You may have a notice that there is a lack of security controls. Maybe they were supposed to be filtering to a device. And yet you're still able to access certain port numbers on that machine.

That means that your firewall may not be configured properly. Or maybe there's no antivirus or anti-malware running on that device. Maybe it's something like a simple misconfiguration. Maybe somebody meant to configure a share, but they didn't assign the right permissions to that share. Now you have access to those through the scanner.

Or maybe somebody enabled guest access. And normally guest access should be turned off. And of course, the scanner may find some true vulnerabilities with an application or an operating system. Especially if you haven't patched lately, you'll find a lot of the new vulnerabilities, and these scanners will give you a notice that it's time to update your system.

If you are going through the results of your vulnerability scan and you notice some of the information is not quite correct, you may have run into a false positive. False positive is when you have a scanner identify a vulnerability. But in reality, that device truly is not vulnerable to that particular issue.

A false positive is something that absolutely does not exist on this computer. That's a little different than a vulnerability that has a low severity. A low severity might be something like an open port number. Well, obviously, every open port number may not necessarily

be a big problem. But it's still an open port number. It still exists. It's just at a lower priority or a lower severity than perhaps something like a buffer overflow in an operating system.

So don't confuse a low severity type of problem with an actual false positive. The reverse of this is a false negative. That's when a device does have a vulnerability but you ran your virus scan and you ran your vulnerability scan and nothing was identified as being a vulnerability. This is almost worse than a false positive.

A false positive, at least we can look at the machine and determine that that was incorrect. But a false negative is something that we'll never know is there because we've scanned and got no results from it. The goal with both false positives and false negatives is to make sure that you update to the latest signatures. These scanners can only scan for what they know about. And the signature update is a critical part for ensuring we're able to see as much as possible on those devices.

And ultimately, you may need to talk to the manufacture of your vulnerability scanner and let them know what you're seeing. They may not have seen the type of environment you're running. And they'd be able to create a signature that's able to solve either the false positive or the false negative issue in your environment. And at the same time, you're probably going to be helping everybody else who has a similar configuration on their network.

**Tags:** active, certification, comptia, false negative, false  
positive, passive, security, vulnerability scan

**Category:** CompTIA Security+ SY0-401

### **Fuzzing – CompTIA Security+ SY0-401: 4.1**

A fuzzing technique will try every possible type of random input to an application to see if a security problem can be found with the application's programming. In this video, you'll learn how fuzzing works and I'll demonstrate fuzzing with an application in my lab.

**Fuzzing** is a very interesting vulnerability testing and application testing technique. It's one that you may hear referred to as fault-injecting, robustness testing, syntax testing, or negative testing. But they all mean the same thing.

They all mean that we're going to send a random amount of data into an application. Maybe an application is expecting a field where you might put your name into the application. Maybe you put anything but a name. Maybe you put all kinds of different information into that field to see how the application is going to respond.

You're looking for the application to have some type of exception. Maybe you crash the application on the client side. Maybe you find that the server crashes. You're going to find something that's a little bit different than what you might expect.

The idea is that if you can find an exception, you're finding a way that this input is not being validated by the application, that may give you additional access to do other things. Maybe you can create a denial of service with that. Maybe you can create a way to see information in the database that the application would normally never provide to you.

This technique is one that was created in 1988. There was a class project at the University of Wisconsin. Professor Barton Miller came up with this project, the "**Operating System Utility Program Reliability.**"

And this was really the **Fuzz Generator**. This was the first time people really sat down and put together a project that would really take an application and put it through its paces. Specifically, in this case, it was operating systems and the applications that were running

an operating systems. And that's where fuzzing was born. And we were able from there to create an entirely new set of testing that we could do with our applications.

Since then there have been many new frameworks created for fuzzing that focus on certain application types or certain platforms or even certain operating systems. We can even see that when we're working with these frameworks that they are very time consuming to use. They're very resource heavy. You have to go through a lot of iterations, because usually your application isn't going to break.

So you have to try a lot of randomization. You have to try different things on different fields. Maybe there's more than one way to put information into an application. We might want to try every single method to get information into that app.

The fuzzing engines— many of them— have a set of tests that are high probability. There are certain things that applications tend to do very badly at. A lot of these fuzzing engines have been built to try to take advantage of those first. So if you can get some of the high probability problems with fuzzing, then you may be able to save yourself a lot of time.

There is one that you can download and try. This is from the **Carnegie Mellon Computer Emergency Response Team, CERT team**. It's called the **CERT Basic Fuzzing Framework or BFF**. And they even have a VMware image you can run from [cert.org/download/BFF/](http://cert.org/download/BFF/) to try some fuzzing on an operating system yourself.

This basic fuzzing framework virtual machine is set up with an application inside of it that is already susceptible to a number of fuzzing techniques. And if we just run this window— we're going to run this application, run this virtual machine— it's going to launch and automatically start the fuzzing process. And it's going to keep trying different fuzzing techniques to try to find a place where the application fails, stops talking, it breaks in some way.

Once the operating system begins, what you'll start to see is the fuzzing process start on the right side. And then you're going to see on the left side any opportunities or any places where the fuzzing actually fails. And it will give you the process ID of when it starts to fail.

So as it goes through all of the different fuzzing, you'll start to see on the left side a lot of different pieces start to load. There we go. There's another opportunity where the application broke.

All of this information is logged. And you can go back into the log. This framework even comes with a number of reports that you can run to see exactly what was given to the application to cause this particular problem.

And from there, of course, you would drill down further and try to take advantage of some of those fuzzy techniques. Obviously, not a simple thing to perform. And as you can see, very time consuming. But it's a great way to determine how secure your applications might be.

**Tags:** application, certification, comptia, fuzzing, security

**Category:** CompTIA Security+ SY0-401

## **Secure Coding Concepts – CompTIA Security+ SY0-401: 4.1**

An application is only as secure as its programming. In this video, you'll learn about security coding, validating input, cross-site scripting concerns, and how to handle exceptions.

If you're developing or writing your own software there are a number of concepts you have to keep in mind to make sure that your code is going to be secure. One of the challenges you have, though, is the process and the time that it takes to make that secure is going to extend your development cycle. So you often have this balancing act between speed and security.

When you're working with your code, and you want to be sure it is as secure as possible, there's usually a quality assurance process. You have somebody test your code to make sure that it's going to be secure. And it's very often not just one type of test, it's many types of tests that you run through just to make sure that you're doing the right things to keep your code as secure as possible.

Eventually, in most software, we're going to find a vulnerability somewhere. It's going to be with the software. It's going to be what the platform that it's on. And the bad guys, when they find that specific problem or that specific vulnerability, they're going to take advantage of it. So we want to be sure, from the very beginning, we're writing code that is as secure as possible.

A very common development process is one where we validate the input that is going into our systems. We determine what the expected input might be. And we compare it with what we get. If we're asking someone for name, if we're asking someone for an address, maybe a serial number, there's probably an exact string or set of strings that we might expect. And we want to be sure that if somebody inputs data that we really do check the data, just to make sure it's exactly what we were expecting.

We want to go through our entire application, document every possible input method, look at forms, look at fields, look at the types of information we're putting in there, and we want to check all of our input. If we have a field for a zip code we know that zip code should only be a certain length. We know that in certain countries there should only be certain characters, and they should only occur as the third or fourth character in the entire zip code.

We should perform exactly all of those checks. We should leave nothing to chance. Because if we end up having a validation problem the bad guys may find a way into our application.

And as we also saw in one of our other videos, you can use fuzzers to try to manipulate that input, to try to throw a lot of random data at your application just to see what's going to happen. Is your application going to have a fault? Is there going to be an error? Is the application going to stop running?

There's all kinds of interesting input and results you can get from a fuzzer. So if you have the flexibility to do that, that's a great way to validate that input.

In our video on cross-site scripting we talked about and demonstrated how painful it can be if one of your applications happens to have an embedded script in there, and it starts giving back information to the bad guys about what people are doing with that app. So of course we want to check for embedded scripts. We want to validate the input for those, just to make sure there's not something in there we weren't expecting.

We also want to check for cross-site request forgeries. We did some of this as well, where we were able just to have— with a one click attack— be able to see the session and identify the session IDs, and then use those session IDs and essentially ride on top of someone's existing sessions and use their same authentication methods to be able to grab that application and see information in there. So make sure that your session IDs and the methodology we use to authenticate people is encrypted and protected in our applications.

Another important consideration when coding is making sure that we have certain routines in play should an error occur. We can't possibly plan for everything, and we should always have something that is a generic message that appears when a problem happens. There should be a graceful process. You shouldn't just get a standard error that pops up as part of the compiler or the script language that you're using.

So if you lose a network connection, the server hangs, a database suddenly is not available to you, should have a certain message— or at least a generic message— that pops up so that you're able to understand what's going on. There are mishandled exceptions, like this one here, where you see the application itself failed and you've got a generic **Microsoft Visual C++ Runtime** library.

So now I happen to know what application was used to develop this app. I know what coding system was used there. And I may be able to take advantage of that with some of my vulnerabilities.

You want to be sure to get rid of any of those default messages. Make it something very generic, or something that is in your pop up screen, so that the bad guys cannot understand what that underlying architecture might be. Use some of those scripting methods, use some of those coding methods, and your software and your applications that you develop are going to be as secure as possible.

**Tags:** application, certification, coding, comptia, cross-site scripting, error handling, input, security, validation, xsrif, XSS

**Category:** CompTIA Security+ SY0-40

### **Application Configuration Baselining and Hardening – CompTIA Security+ SY0-401: 4.1**

You must first determine a baseline for application security before you can begin the process of hardening the technology. In this video, you'll learn some best practices for security baselining and some techniques for hardening the operating system and application environment.

There are many different aspects to securing an application. And that's because an application has so many different components associated with it. So an important thing to do is to identify all of those different components, and understand how the application is used by each one of those.

So look at the browser that's being used by the application. Look at the operating system it's running on. Does it have any service packs, or any security patches that need to be associated with it?

You need to understand exactly the way the application is running. Because if any one of those things changes then we need to be aware of it, and understand how that impacts the security of the application. And of course, you have so many applications to choose from. You have to do this for every single one that you have. And they're all going to change over time.

There's going to be updates to the operating system. There's going to be updates to the browser. Maybe someone else uses a different browser. Is that an appropriate browser to use for this application? You as the security professional have to make that decision.

The **baseline** is going to be updated constantly by, not only security patches and normal operating system patches, the application itself can change. So of course you have to keep track of that. And then the other applications on the same workstation and on the same server are going to change. So you also have to keep track of those. Especially on servers— if a server happens to be taken over by a bad guy using a different app, they may possibly have access to this other application and be able to share information between them.

After you do a major update, after you do a major change to your workstations, make sure that you do another baseline. Make sure you understand the impact that's going to have. Make sure that the system remains secure by adding these additional patches. You don't want to get in a situation where you've updated an operating system, you've added a new patch, and unfortunately you've opened up that operating system in the application to other types of vulnerabilities.

We've talked about hardening operating systems. But you also have to think about hardening the applications as well. And a number of the best practices for operating systems still apply for these applications. For instance, we want to make sure that the operating system this application is running on is secure as possible. So make sure that we have the latest security patches, make sure that we have the latest service packs, so that nobody can get into the operating system and perhaps gain control of that application.

The application itself is going to have updates. You're going to get updates from the manufacturer. You're going to update the application if you wrote it in house. There will be changes associated with that.

These changes may bring new features to the application. They may be bug fixes. And for everything that changes, we need to have an understanding of how that impacts the security of the application itself.

You all should also use the best practice of the least privilege access. You don't want the application to have read access to an area that it should not have read access. You don't want it to be able to delete files that perhaps it should not be able to delete. And normally these file access, and the ability to delete files, isn't a problem until somebody gains access to the application who shouldn't have that access, and finds ways to have the application delete files for us.

If the application itself never had access to delete files then that's one thing you'll never have to worry about. By setting those least privilege policies now you can be assured that, should something odd happen— a bad guy get hold of the application, be able to manipulate the app, or perhaps the application just have a bug— it's not going to cause a problem for other people.

In many environments you'll even see the application machines, these workstations, are very tightened down. You can't change the background on your desktop. You can't install new applications. You can't change colors on the screens. You can't change fonts because the security administrators know that every single one of these changes could impact the stability and the security the operating system.

So by hardening the system, and hardening the way the application is going to work, you're going to have a much safer environment for everyone.

**Tags:** application, baseline, certification, comptia, configuration, hardening, security

**Category:** CompTIA Security+ SY0-401

### **Application Patch Management – CompTIA Security+ SY0-401: 4.1**

A patching strategy for an application should be well designed. In this video, you'll learn about application patch management, how different operating systems are patched, and some of the challenges with maintaining a well patched computing environment.

One of the constant things that you will find is that your applications will always need to be updated. There are so many different ways to go about doing this.

But we also have to think about why we would want to update our applications. One is to get some additional features. A new version comes out. It's got a most requested capability. It has some features that are going to add value to the way that you use that application. So it makes perfect sense to update your application that way.

Occasionally, you will find bugs. You'll find problems with the way that the application works. And so the people who develop the application will fix those bugs and provide you with an updated version that fixes all of those problems.

We also need to think about, of course, security. If you find a security vulnerability in an application, it obviously is going to be very important to have that application to the latest version. You don't want to have the bad guys discover that you're using this application that has this known vulnerability. Because they're going to attack it. They're going to exploit that vulnerability so that they can use the application for their own purposes.

These application updates will often come in through the operating system updater itself. In Windows, for instance, there's the capability of Windows Update, which provides you with the front end to update not only your operating system, but the applications that you're running in your operating system, if those applications are from Microsoft. It's a very simple way to have this work. It's in many cases automated. And it is an individual workstation by workstation method. You would use your Windows update on a machine to update the local operating system for that.

If you're in a larger environment, though, that could be a bit tedious. You don't want 1,000 different computers all going out to Microsoft, all downloading exactly the same patches. That's going to use up a lot of your internet bandwidth. So what many administrators prefer doing is using the Windows Server update services. And that Windows Server update services centralizes in the Patch Manager that's a server in your organization. The patch server downloads the patches, and then it provides the patches to the end user workstations.

Obviously, you have to configure every workstation to be able to use that centralized patch management server. But in the Windows environment, that's done through group policies. It's an automated process. When you're in a very large domain, it's not as hard as it actually sounds. And once you have that server set up, it's very efficient. And you have control over exactly what patches are pushed out and when they are pushed out to the end users. If you're in **Mac OS X**, then you have the Apple menu's software update option. So you can update the software on an individual basis in that OS.

Every operating system has these— **Linux**, of course. You can use many different ways to do this in Linux. And it depends exactly on the distribution. You can use `rpms` or `yum` or `apt-get`. If you're running this in a **GUI**, you can run a software update in the **GUI** itself so that you can process it that way. Regardless of the operating system, you're going to want to use whatever method is available to you to make sure you keep the operating system and your applications up to date.

Keeping your systems up to date is not the easiest thing. It takes a lot of management. It takes a lot of study to understand what patches are out there.

Let me give you a good idea of this. This is a Windows 7 machine that I've not patched in quite some time. I'm going to go to my Start menu, right to the Control Panel. And as we mentioned, Windows has a Windows Update feature that I can choose. This Windows Update is going to show me exactly what updates are available. There are 28 important updates available, and 52 optional updates. Now you have to make a decision. Of these 80 different updates that are out here, which ones do we put on our computer?

If I look at the important updates, you'll see a big list come up. I have no idea what these are. In fact, you'll see a lot of them are called Security Update for Windows 7. And there's no other explanation of this, other than the knowledge base article and the information to the side that tells us about it. Now fortunately, there's more information. We can always click More Details. It will open up a browser and give us more information on Microsoft site on what that specific patch is going to provide for us.

And we as the security people now have to decide is this a patch that makes sense to deploy on our workstations? Is this patch going to break one of our critical applications? Very often, we get these updates from Microsoft. We now have to go test them. Let's take our test machine in our lab. Let's load up these patches. Let's run the apps that we run internally, and make sure that it's not going to break anything. And then we're going to deploy them.

Obviously, that takes time. And when you have a vulnerable operating system, you don't want to keep it vulnerable for very long. So there's a balancing act between making sure that our systems continue to operate and keeping them secure. And that's, from a security perspective, something we always have to keep in mind.

Once you've tested the patch, you're sure that it's going to work on your system, it's not going to break any other applications, and you're able to deploy it, you then have to go back and reconsider doing another application-level baseline. Now that our system is using a new set of patches and it's at a different security level, we want to be sure that we understand exactly what's included with those patches. So that if we need to rebuild another system, or we need to check the security of our app, we'll know exactly the operating system that it's running on.

**Tags:** application, certification, comptia, patch, security

**Category:** CompTIA Security+ SY0-401

## **SQL and NoSQL Databases – CompTIA Security+ SY0-401: 4.1**

Our databases are the core of our applications. In this video, you'll learn about SQL and NoSQL database technologies and the type of data we would store in each type of database.

Every organization has data. There is generally a database where you're going to keep all of this information. And one of the most common database types is a **SQL database**, or **S-Q-L** database you may hear it called. It stands for **Structured Query Language**. It's a standard way of getting information and storing it in one central place.

All of this information then is centralized in, usually, a single database. But it can be in multiple databases as well. And one of the key pieces of these **SQL databases** is that it's very easy to request information from the database, and very easy to retrieve information from the database. And when you're storing a lot of information, you really are trying to find the best way to gather the information that you're going to need.

This information is stored in a structure called a **Relational Database Management System, or RDBMS**. These databases are big, flat fields of information. They look very much like a spreadsheet.

There's a lot of information stored in a table. There are rows and columns. If you were to visualize it on the screen, and show that information, it displays very much like a spreadsheet.

You might have multiple databases where customer records are in one database. And then you might have customer orders in another database. And then there's usually a field between both of those that matches, or identifies, the owner so that you can then look up that information and compare information across separate databases.

And it's very fast and very functional. And it's so much of a standard that a lot of applications that you can buy off the shelf will automatically know how to communicate and gather information from your database, as long as it's a SQL compliant database.

One of the advantages of **Structured Query Language** is that it's everywhere. Practically every organization is using a **SQL-type database** to be able to store their data. If you're a Microsoft shop you probably have **Microsoft SQL Server**. If you have a lot of **Linux servers**, you're probably running **MySQL Azure database**.

Both of these have the Structured Query Language. But obviously the engines themselves are different between the **MySQL** and the **Microsoft SQL Server**.

There are a lot of huge advantages for using **SQL databases** when you have this structured kind of data. But what if your data is not so structured? And in those particular cases we want to go beyond the realm of a **SQL-type database**.

These newest kinds of databases are called **NoSQL databases**. And the name sounds like it's the anti SQL, but that's not the case at all. It really stands for **Not Only SQL** information, which means you could store formatted structured SQL information within this database. You can also store completely non-relational, non-associated data within this database as well.

Again, that's the concept behind big data, is we want to store as much information as possible. And we're going to sift through that data later on to see if there's any relationships between all of this very diverse data.

The idea behind big data is that you're going to have a lot of information. You're pulling it from so many diverse sources, and storing it in a database, in usually over a long period of time. So the **NoSQL database** has to be able to scale to these larger amounts.

You're going to have extremely large data sets. These data sets are going to be completely unstructured. And it's really going to be up to you later on to get this big data, and somehow find the relationships between all of that information.

Sometimes there's relationships between the data, sometimes there isn't. One of the challenges with big data is finding those relationships. And these databases, especially NoSQL databases, give us the flexibility to store as much information as possible. Later on you can go through and try to determine where the different pieces of information are that you're going to need to gather.

The goal is, immediately, is just store it somewhere. And the NoSQL databases are specifically designed to do exactly that.

**Tags:** [application](#), [certification](#), [comptia](#), [database](#), [nosql](#), [security](#), [sql](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Server-side vs. Client-side Validation – CompTIA Security+ SY0-401: 4.1**

Should you validate input data on the server, or on the client? In this video you'll learn the advantages and disadvantages of using both of these validation strategies.

One of the ways the bad guys like to get into your systems is they try to give it information that the system or the application is not expecting. And if they can do that in a way that they can predict the outcome, they may be able to gain access to data or systems that normally they would not have access to.

To try to combat this, what we want to do is to validate all of the input that the end user may be making to that application or into that service. One of the ways to do this is on the server side. The end user is typing in information and they're sending it to the server to be evaluated. Maybe they're sending a sequel call. Maybe they're registering on a website or they're looking for search information. And what we'll do is grab that registration information or have a look at that search query to determine if this is really a valid query or valid input, and we do that once the information has been sent to this server when you're doing a server-side validation.

This is really going to help against those malicious users who are trying to take advantage of some shortcomings in your programming or they're trying to find a way around the interface of your server, and by checking it on the server side, you can make sure that you've got the final step before anybody is able to get to the data or get to the system. We could, of course, do this validation at the user side before it's ever sent in to the server. So if the user has an application that they're using or they simply have a browser front end to this application, we would have some intelligence built into that application that checks the query first and if all of the query information looks OK, everything seems to match up, it's not trying to take advantage of any vulnerabilities, we'll then send that query off to the server.

One of the challenges with this is that it could filter legitimate input from the user. You have to be very careful about how you design those validation checks especially if it's all being done at the client side. If you make a mistake with those validation checks, you're then going to have to update all of your users' applications so that you don't make that mistake in the future. This can also provide some additional speed since you're doing these checks down on the user side where generally you have more computing resources

available. On the server side, the servers are very busy doing a lot of things, and if you're doing validation checks on the server, it may take additional time.

You really want to use both server-side and client-side validation, so you want to validate as much as possible on the client side. But some clients are smart. The bad guys will use a third-party application that looks and feels like the original app and sends information to the server, so they can circumvent any client-side validation you may have in place. So for that reason, you want to also have your server-side validation so that once the data arrives at the server, you can perform the normal checks and make sure that nothing odd is going on with the information being sent in to your server.

**Tags:** [application](#), [certification](#), [client-side](#), [comptia](#), [data](#), [security](#), [server-side](#), [validation](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Mobile Device Security – CompTIA Security+ SY0-401: 4.2**

Mobile devices have completely changed our perspective of network security. In this video, you'll learn about the important aspects of securing a device that is constantly moving and outside the direct control of your organization.

We are obviously now a mobile workforce. Everywhere we go, we have our mobile devices with us. And everything we do during the day seems to go back to our mobile devices. Our companies can sometimes provide these mobile devices for us. And occasionally we'll have our own that we bring into the office as well. You'll see these referred to as **Bring Your Own Devices, or BYOD**.

So your organization has a bit of a challenge. You not only have company assets— you need to secure the data and the resources on those assets— but you now need to do the same security to people's own personal devices. And you generally accomplish this through something called **Mobile Device Management, or MDM**.

This is a centralized manager. It usually consists of some very intelligent software usually running on a server or specialized hardware. And it's able to communicate out to the internet and out to all of your mobile devices wherever they happen to be in the world. This is obviously a very specialized functionality.

This isn't like our old firewalls or intrusion prevention systems. This is an entirely new category of security and management that's specifically designed for these mobile devices. The mobile device management software allows you to do a lot of things on these mobile devices. And usually, you're setting policies that will affect all of these devices.

You might want to decide what applications are appropriate to run on that device. Maybe you want to enable or disable the camera functionality. Maybe you want to be able to control everything about that mobile device. And you can do it all from these mobile device managers.

Some of these mobile device managers allow you to partition off a separate section inside of the mobile device that's just for the company information and the company control. And then it will still allow you to use the rest of your mobile device as a personal device that you can take your personal camera pictures, but not use that when you're using the company side of that mobile device.

There's a lot of functionality that's enabled when you start using a mobile device manager. You can push out policies that requires that all of the mobile devices have a certain access code set. Maybe it requires a lock screen PIN so that you can be assured that your mobile

devices with your company information just aren't sitting around and accessible. You can tighten them down and make them as secure as you'd like.

We're generally storing a lot of information on our mobile devices. It might be data from your organization. It might be contact lists. But all of that information is proprietary and private to your company. So a lot of organizations will encrypt the data on all of these mobile devices. That way, if the device is lost or misplaced, at least you know that the data will not be accessible to third parties.

There's a lot of ways to implement this type of encryption on a mobile device. This screenshot from an Android operating system shows you that you can enable encryption in memory. And you can set the memory strength to be strongest, stronger, or strong. This gives you some different levels that you can use for encrypting and protecting that data.

And you may be asking, well, why would you not want all of your data to be encrypted at the strongest possible encryption? Well, that's because as you have stronger encryption, it requires more resources of the mobile device. There will be more CPU cycles used. Therefore, more battery will be used, and more memory. And of course, on these mobile devices, we have a finite amount of all of those resources.

So you want to set the encryption to the mode that makes sense for your company but still is going to allow you the most functionality of the mobile device. One important consideration is that if you're going to encrypt all this data, do not forget your password. This is where you have access to a certificate that is going to be used to encrypt this data. And if you forget a passphrase or password that allows that access, then all of that data is now going to be inaccessible to you and everyone else.

If you've ever lost a mobile device, one of the things that concerns you immediately is that someone is now going to have access to all of your applications and all of your data. And one of the challenges then is how do you make sure that none of this information is going to get into the wrong hands?

Well fortunately, all of these remote devices these days have a remote wipe functionality. This will completely sanitize the device. And you can generally do this from your mobile device manager or from a browser front end. If you're using an Android or an Apple device, it's very simple to go into their front end and choose to completely remote wipe the device. This is the remote wipe screen from my iPhone that tells me that I can erase everything on this device all from a browser screen.

I've logged in and chosen to do a remote wipe. And it tells me, this will permanently delete everything on your mobile device. Once wiped, your iPhone will no longer be able to display messages. You won't know where it is. It's completely factory reset. And they even have a check box here that says, I understand I cannot undo or stop this action.

But if somebody now has access to your mobile device, maybe a remote wipe is really the best thing to do at the time. But you need to plan for this now. You need to connect your device to a mobile device manager or make sure that you have arranged for your Android or Apple device to have this remote wipe functionality. If you've not configured it beforehand, it's going to be very difficult then to have abilities to remote wipe this data later on.

All mobile devices these days have a screen lock functionality. So when you're not using your phone, it automatically times out and goes into a locked mode. And you have to input the unlock key to then gain access back to the mobile device. This could be something very simple, like a four digit passcode. Or maybe you can create a very strong passcode that includes both upper and lowercase letters. So you've got a lot of options for how you want to define what that lock code ultimately is.

And you can also define what happens if you try over and over and over again, and you still aren't able to use the right lock code. Somebody gains access to your device, you can set it up on iOS to erase all data on this iPhone after 10 failed password attempts. That way, if somebody does gain access, and they're trying to type in information just to guess what your passcode might be, after the 10th access attempt, and it's incorrect, your entire phone is now going to be wiped.

So you do want to make sure that you always have a good backup if you're going to enable that functionality. And that does speak to how you should define these lockout policies. If you're managing a lot of different devices, you'll probably want to have a set of global policies configured on your mobile device manager. It might have some very aggressive lock-out timers. If nothing happens for 60 seconds, lock the phone, or five minutes, or 10 minutes.

And at that point, you want to define what's going to happen when this phone is locked, what type of passcode needs to be inputted to unlock the phone. And if you do have a situation where somebody's trying over and over to brute force that passcode, you need to define what happens at that point.

If you ever have lost your phone or your tablet, you know there is some great GPS functionality built right into the technology. You can get very, very precise tracking information that will get you back to that device all within a few feet of each other. These things can be very useful for your organization so that if somebody does lose their phone, you can redirect them to where it might be. Or you could at least know where people are in the organization at any time during the day.

But they can also be used for bad reasons. If somebody wanted to track where you were going and what you were doing, this would be a very, very good way to do it. Most devices will give you the option to enable or disable that functionality. If this is a company owned device or it's a BYOD where you have brought your own device to the company, you may not have a choice. It may be on all the time just so the organization knows where their applications are and where their data is all the time.

These mobile device managers are incredibly powerful. They really can control every aspect of your mobile device. And they can even control what applications are loaded and what applications can run on your device. And your mobile device manager administrator can now set policies and define exactly the types of apps that are allowed on your device.

If there is an unapproved app, you can restrict the access or just remove it completely from the mobile device. These **MDMs** are extremely powerful, and they do have complete control over your devices. The more advanced mobile device managers will allow you to segment off a certain section of your mobile device that's just for corporate data. So you can store your data and applications and control what users do in this partitioned area, but still allow personal use of the mobile device.

Some of our phones and tablets have a slot for removable storage. So you can plug in some storage, copy information to that storage device, and then remove it. Obviously, when it's removed, the mobile device manager has no idea where that removable data is. So one of the things you can configure in your mobile device manager is to allow or disallow someone to write certain kinds of data to that removable memory.

Some of the other features of the phone can also be enabled and disabled. You can enable and disable Bluetooth. If your organization is concerned about using this over Wi-Fi, you can disable Wi-Fi, or disable the camera functionality. Every little piece of that hardware and software inside of that mobile device can always be managed from that mobile device manager.

If you're like me and you have a hard time just trying to find your car keys, imagine managing hundreds or thousands of mobile devices on your mobile device manager and wondering where all of those devices might be. And in large organizations, these devices might be anywhere in the world. So it's very useful to have some functionality to be able to track the assets and know exactly what inventory is out there being used in the field.

Some organizations require you to buy your own phone and to buy your own tablet, and then they don't have to worry a lot about the asset itself. All they have to do is control what happens on the asset. But of course, if the organization is buying the hardware, then they're obviously going to need a way to keep track of where that asset happens to be. That's why location services on these mobile devices are very important, so that the mobile device can check in with the mobile device manager and tell the mobile device manager where it happens to be in the world.

And of course, if you're on a plane or you're somewhere where you don't have a GPS signal, at least the mobile device manager will know the last time that particular device was seen out there in the world. Some security policies will include what things will be monitored on your mobile device. There's obviously privacy concerns. And in different countries, there are going to be different rules and regulations about what is private and what is not private. So your mobile device managers have to be smart enough to know that in Germany, there's a completely different set of privacy concerns than there might be inside the United States.

**Tags:** certification, comptia, encryption, gps, inventory, mdm, mobile, remote wipe, screen lock, security, segmentation

**Category:** CompTIA Security+ SY0-401

### **Mobile Application Security – CompTIA Security+ SY0-401: 4.2**

Managing the applications running on mobile devices requires some additional security planning. In this video, you'll learn about key management, managing credentials, geo-tagging, and application whitelisting.

Our mobile devices store a lot of data, and if we want to protect that data then we're going to need to encrypt that information. Fortunately, there's a lot of encryption technology already within your mobile device, and being able to store that information becomes even more important as that mobile device moves around the world. You want to be sure that nobody gets their hands on your private information.

Many times you can store this data encrypted on the mobile device. So as soon as you see it on the screen, you store it to the device and immediately it's going to be encrypted. Very often, the memory itself will allow you to encrypt the data, and you have to have the proper credentials to get into that mobile device to be able to gain access to the data. Of course, we're going to be sending that data across a network. It's going to be a wireless network, a Bluetooth network, the mobile provider's network.

So there are encryption and security APIs— these are application programming interfaces— that send this data across the network via SSL, which is obviously a very popular and ubiquitous encryption technology. Usually if you're communicating back to a mobile device manager, you will need to set up that mobile device manager with the proper SSL certificates.

You'll need to have something like a trusted certificate authority or have your own certificate authority in your organization that then is pushed down to your mobile device, so that the mobile device will then trust the device that it's communicating with. It's usually your mobile device manager administrator who's setting up these policies for encryption, so they'll be sure that your mobile device will encrypt data on the device as well as encrypt the data as it's going across the network.

Many of the applications on our mobile devices require that you log in with a user name, password, or some other type of authentication mechanisms. This is usually something that's separate from the application code itself that someone is writing, and it may be integrated into the mobile device itself. These are almost always server based, so your credentials— the user name and the password— are often stored on that remote device. That way it's very easy to manage the user name and the password and whatever credentials you're using.

If you had all of those credentials running on all of your different devices, the administration of those may be a little bit more difficult to manage. These are often communicated across the network in an encrypted form, so SSL is commonly used to be able to communicate out most common wireless networks or mobile device networks.

Sometimes the application doesn't actually do any encryption, and that's a problem if we're sending information out over the network. So it's very common for a mobile device manager administrator to do some auditing of the applications to make sure that when information is sent from one end of the network to the other, that all of that information is going to be encrypted and completely protected. It's also common, both on our desktops and our mobile devices, to use a third party encryption mechanism to gain access to an application.

One very common one is you'll see a button to log in with your Facebook credentials, or log in with your Google credentials. Those are using something called a transitive trust. If you're authenticating properly with Google, we can trust then, therefore, that you are that user and we're going to allow you then access to the application that you need to use.

The location services functionality on our phones and tablets always know where we are. And it uses a number of different mechanisms to be able to narrow down where you happen to be. Might be a GPS. It might be using the wireless network you're communicating with. Or it may triangulate where you are based on your mobile provider's antennas. In any case, your device is going to keep track of everywhere you go and it's going to add this device location information to the metadata of the documents you create.

If you take a picture, if you store a document, the location of where you created that picture and where you saved that document is going to be included with the information that is sent and stored on your device. This means that anybody who receives that document can look through the metadata and determine the longitude and latitude of where you happen to be. This is very easy to track. It's not encrypted. It's simply included in plain text with the document that you happen to be sending.

So this can obviously have some security concerns associated with it, especially if you're not interested in telling people where you happen to be. If you upload a picture to social media, the social media will often determine where you are and not only post your picture, but tell everyone where this picture was made by looking at the longitude and the latitude. So if you're trying to maintain a level of privacy and not let people know your exact location, you may want to see about changing a number of these location services on your mobile device.

Our phones and our tablets are mobile computers. Extremely powerful technology. And we usually have a number of different applications that we run on these mobile devices. Some of them are games. Some are business applications. But all of them have to be loaded and run on that mobile device. The challenge, of course, from a security perspective is that not all of these applications are secure. Some are absolutely malicious and are designed to gather as much data as possible and send that off to the bad guys.

Android malware specifically is a growing concern and there has to be a balancing act between what you want to allow your users to run and also keep all of their data safe at the same time. Many mobile device manager administrators will address this challenge

by creating application whitelists. They will have a list of the applications that are allowed to be installed and run on that mobile device. Every other application is therefore not allowed on those mobile devices.

This obviously requires a bit of administration by whoever's using the MDM, because every time a new application needs to be included, it then would have to be added to this whitelist. And that may be a good trade-off for your organization. You might not mind adding new applications to the whitelist if you can be assured that it's going to protect both your user's data and your company data from anything malicious on that mobile device.

**Tags:** [certification](#), [comptia](#), [credentials](#), [encryption](#), [geo-tagging](#), [key](#), [mobile](#), [security](#), [whitelisting](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Mobile BYOD Concerns – CompTIA Security+ SY0-401: 4.2**

The complexity of mobile device security is compounded when the user community takes advantage of “bring your own device.” In this video, you'll learn some security best practices around securing these user-owned mobile devices.

From an IT security perspective, the goal of **BYOD** is one that has a number of advantages and a number of disadvantages. This concept of bringing your own device to work means that the organization doesn't have to spend a lot of money purchasing more devices when we all have a phone, and we all have a tablet anyway. The challenge, of course, is that the security team still needs control of that device. So there is a balancing act of having the right amount of control, yet still allowing you to use your personal device.

In almost every case, an organization is going to use a **Mobile Device Manager**, or an **MDM**. That means that they're going to connect your mobile device to the MDM. And at that point, they'll be able to deploy the policies that are able to keep that device secure wherever it happens to go. There's probably going to be a completely different **AUP**— that stands for an **acceptable use policy**— for these BYOD devices. There's obviously acceptable use for the computers that you're using inside the walls of the building where you work.

But what are the acceptable uses of a device that is personal to you that you use outside of work? Which policy is going to win— the personal policy or the business policy? And all of these need to be well-defined and communicated to everyone in the organization. It would be great if our technology was one that allowed us to do whatever we'd like. But from a mobile device perspective, there are a lot of different and very proprietary environments out there, such as **Android**, **iOS**, **Windows Phone**, **BlackBerry**, and there's others as well. So where do you draw the line? We need to make sure that we're able to keep all of these devices secure. But we can't manage every single one of these proprietary platforms. We're going to have to get a Mobile Device Manager that is then able to access and manage whatever we've chosen to be of these approved devices.

We might say in our organization we're going to allow anything that's iOS and anything that's Android. And of those devices, we're going to manage them through our MDM. We're going to need to then purchase the Mobile Device Manager software. There may be even a hardware component associated with it. We're going to have to, of course, get trained on this technology. And there's probably going to be ongoing maintenance costs as well. This device is going to need access to the internet— which ultimately, will allow you to communicate with all of those mobile devices wherever they happen to be in the world.

So now you've started working for this company. Your mobile phone is now part of the Mobile Device Manager on the network. And now the organization is in charge of

supporting that device. If you lose your phone, your first call is probably going to be to your corporate help desk to let them know to lock everything down, or even erase everything that happens to be on that phone. It's not generally going to be to your wireless provider. That's probably your second call. But obviously, your corporate information is going to be the most important thing to secure if that particular mobile device gets out of your hands.

The corporate office is then probably going to wipe the data. You're obviously going to need some backups if that happens. Or maybe there is a partitioned area of that mobile device. And the Mobile Device Manager will simply delete everything in the partitioned area, leaving your mobile device absolutely intact with all of your applications and all of your data.

This gets to be a little complicated, as you come onboard and offboard in an organization. If you start with a company, they'll obviously connect your device. But what if you leave the organization? They're obviously going to want to either delete everything on that device, or perhaps just remove that device from the Mobile Device Manager. If you wanted to be very secure with your personal information, it might even make more sense for you to do a factory default wipe of the device, and then reinstall from a backup. That way you can be absolutely sure that there are still no lingering connections back to that corporate environment.

These mobile devices we carry around are little computers. They're powerful pieces of technology that allow us to do a lot of different things. So we're installing applications all the time. And we're moving data around on this device all the time. And every time we install something new, we have the potential to introduce something malicious into this little mobile computer that we're carrying around.

Sometimes when we will patch a mobile device, we'll break something or create a security problem, just by adding a patch into this. And we don't want to disable the functionality of our corporate or work applications. So we want to be very careful of the applications in the patches that we install on these devices. It might be a good idea to have an anti-virus or anti-malware application running on our mobile device. If this is integrated with a Mobile Device Manager, the MDM may be able to perform scans of these applications and data, and protect your device from that end.

We have, obviously, technology concerns dealing with security. But they're also policy concerns when you're dealing with these mobile devices, especially if it's a BYOD device where the end user owns the device. And effectively, it's a private device that they're using in their private life as well. So there needs to be well-defined policies that sets up where these lines are drawn. At what point is this a corporate asset, and at what point is it a personal asset?

And everything needs to be well-documented and communicated to everybody who's going to be using these BYOD devices. We spoke earlier of segmentation of this data. Some Mobile Device Managers can partition off a separate section of the mobile device so that you can really define what happens to be the corporate side of that mobile device. And there's a very clear dividing line as to where the private side of that mobile device might be.

You've probably seen in your day-to-day life that there are certain places that have policies restricting the use of the camera that's on your mobile device. For instance, if you're a member of a gym, there's big signs up that ask you not to use the camera when you're inside the gym. This can be a bit of a challenge, of course, from a privacy perspective.

And it's certainly a concern from an industrial espionage perspective. You don't want visitors coming into your building and taking pictures of your documents and the inside of

your facility, if that information needs to remain private. There are some Mobile Device Manager policies that will restrict the use of the camera so that you could apply technology to restrict that. That way you're not relying on the end user to simply not use the camera. You're completely disabling the functionality of that camera or video functionality.

There is also Mobile Device Managers that allow you to do something called geo-fencing. This means that they will recognize when you get to a particular area and when you're inside of that area, a certain policy will apply. For instance, if you're inside of your corporate building, it might automatically disable your camera. And when you get back out to the parking lot, your camera is re-enabled. So you can use it wherever you happen to go.

The legalities regarding the data that's on your mobile devices are different wherever you happen to be in the world. And a lot of these laws are still being created. And we're still trying to determine where do we draw the line, with what information is private and what information is owned by your company. There are also concerns, obviously, if there's been a security attack, if somebody has gathered information from inside of your network. You may want to go through every bit of data on these mobile devices. So there needs to be some policies and procedures that will set limits, or perhaps allow you access to all of the data or some of the data on these mobile devices.

With a desktop inside of your company, you generally have complete access to it. The security person will show up, in some cases, remove the entire machine, replace it with something new so that then they can go back and look through the forensics of that device.

The mobile device, obviously, has personal data inside of it. The forensics process you're running may need to exclude certain aspects of the mobile device to maintain the privacy of that individual. So the question then becomes is who really owns this data? The BYOD devices is owned by an individual. But the data on the device needs to be accessible by the organization. So there needs to be some very specific policies put in place so that everybody knows what information is private and what information is available to the organization.

**Tags:** anti-

virus, aup, byod, camera, certification, comptia, forensics, legal, mobile, ownership, patch, privacy, security, video

**Category:** CompTIA Security+ SY0-401

### **Operating System Security and Settings – CompTIA Security+ SY0-401: 4.3**

An operating system has hundreds of configuration options that can affect the security of the platform. In this video, you'll learn about the customization of user rights, log settings, file permissions, and many more operating system options.

The operating system is an incredibly important starting point for any type of security. If the operating system is vulnerable, you generally will have access to everything that is stored in the operating system, which means all of your applications and all of your data. If you start getting into operating systems and looking at all of the different security settings, 0 you'll run into, literally, hundreds of different settings that you can configure to define what the security posture might be for your operating system. And you're generally going in and only changing a number of these. Maybe you're only focusing on changing firewall settings within the operating system. Maybe there are certain applications you don't want to even have installed in your operating system. And, of course, you want to do some best practices such as disabling guest accounts on your operating system.

Let's categorize what some of these security settings might be in our operating system. User rights allow us to change what a user may have access to inside of the system. They might have access to certain kinds of files, and their access to that file might be read only, or we may allow read and write access. And, of course, this usually includes setting up groups of users so we can create logical groupings. For instance, the shipping and receiving department may have a certain type of access to these files, but the accounting department may have a completely different access to the same files.

Your operating system can log a lot of information. So you may want to define exactly what gets logged in the operating system. You may even want to take that log information and send it out and forward it to a centralized source, so that you can collect logs from all of your operating systems at one time, and then access a report on those from a central reporting console.

In the operating system you can also lock down individual files. There may be operating system files you don't want anyone to have access to, or you may only want to restrict, perhaps, the execution of certain files, but not the modification of those important system files. If we're managing a Windows operating system, we also need to think about what access people have to the registry. Generally, our administrator should have access to the entire registry, but we may want to limit access, especially to sensitive areas that deal with the operation of the OS itself.

And, of course, we can set account policies. We need to define what users can do in the operating system, and what they can't do in the operating system. So all of these hundreds of security settings can be modified and changed to create the correct security profile for your OS.

**Tags:** [certification](#), [comptia](#), [files](#), [firewall](#), [logs](#), [operating system](#), [OS](#), [permissions](#), [registry](#), [rights](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Anti-Malware – CompTIA Security+ SY0-401: 4.3**

How can you keep the bad guys from infecting your computer? In this video, you'll learn about anti-virus, anti-spyware, anti-spam, pop-up blockers, and host-based firewalls.

When we think about malware and all of those different things out there that get onto our computers and create problems for us, certainly viruses is one that's very top of mind. There are so many viruses out there. There are literally thousands and thousands of new viruses every week. There's single viruses and variants of those viruses. It's really just a matter of time until you download a file that happens to have a virus inside of it.

To look at some of these statistics— these are from May 2011— these are computers that were running Kaspersky Lab products. They collect statistics from all of the systems out there that have opted into this. They found that there were over 242 million network attacks blocked for the month.

There were 71 million plus attempted web-born infections prevented, where somebody was downloading an infection from the web. There was 213 million plus malicious programs detected and neutralized on the computer itself. And there were heuristically 84 plus million heuristic verdicts register, which means the heuristic engine found something that it didn't immediately identify as a virus. It found some application acting like a virus and stopped it.

If you want to see all these statistics, they're right here at this link. But it's certainly speaks to how prevalent viruses are in our environment today. You always want to have an

antivirus application installed on your computer. There are plenty of very efficient and easy to install and run applications out there.

There are still people that will come up to me and say, I don't need an antivirus application. I know what to click on. And I know what not to click on.

And unfortunately, it doesn't matter anymore. The bad guys have realized there are ways to get around you clicking on things to have viruses downloaded into your computer. Make sure you install an antivirus app. Make sure those applications are current, that the signatures for that application is current.

And make sure that you still stay on top of this. No single antivirus program can stop 100% of all viruses that are out there in the wild. These things happen just so quickly.

So make sure you keep an eye on what's happening with your computer. If anything looks out of the ordinary, it probably is. And you want to perhaps have more than one antivirus application or at least scanning program on your computer to double check what's out there. It's a very important aspect of making sure that when we download files from somewhere else that they're going to be as safe as possible on our system.

One way to double check viruses on a file you may have downloaded is to use a service like VirusTotal. You can find this at [virustotal.com](http://virustotal.com). And what it will do is take a file that you've uploaded. And it will send it through a number of antivirus scanning engines to see if it happens to get a hit on any one of those.

So I've chosen a file on my hard drive called not a bad application dot exe. And I asked to send that file up to VirusTotal. It's going to take the file, upload it to VirusTotal. In fact, it says that this file has already been submitted by someone else in the past.

It has an MD 5 hash that matches exactly for this file. It was first seen in 2008 and last seen in 2011. So this gives me an idea of what's there.

But maybe I'd like to reanalyze. Maybe I don't trust what somebody else may have uploaded. Or I'm not completely sure. Or I would like to update this with the latest engines to see what they show.

So I'm going to choose to reanalyze that. VirusTotal will look at that. It will queue this particular app. As you can imagine, there are a number of people using this service. And it will show you where you are in the queue. And then it will start going through its virus signatures.

Another thing I like about VirusTotal is you can leave messages in here about what other people have seen with this particular application or with this particular signature of a virus that has been seen. And as it goes through the antivirus, it looks that Bitdefender found a worm. McAfee found Conficker. G Data, Kaspersky— there are all these different antivirus engines, and they're all finding Conficker inside of this executable that I've uploaded.

So I know that if I'm looking for an application that would have done bad things on my computer, this is it. And VirusTotal was able to qualify that for me and see that practically everyone out there is able to identify this worm and this virus. And now I can keep it off of my system.

In the world of malware, we also have to think about spyware. And spyware is generally something that is already installed on your computer that's now watching things that you are doing. And it may be watching you to provide information back to a mothership.

And that information is probably anything from how we're browsing the net. It could also watch for usernames and passwords. It can contain a lot of details about where you might

be logging in or even everything you type in with the key logger, and send all that information to a third party. Obviously spyware, a big problem, because we don't want our private information getting out to other people.

So we want to be able to have an anti-spyware application on our computer to watch for anything strange that might be going on. This is very often integrated into your antivirus engine. And that makes sense, because our antivirus engines are already looking for everything that's running on our computer.

Why not also have it look for spyware that might be on our system as well? There are also standalone anti-spyware applications you can get. And some of those are very, very good that go even a little bit beyond what traditional antivirus programs have been able to do for us.

These pieces of spyware are on our computer to watch what we're doing. So we want to have our applications and our anti-spyware technologies also watching what these apps are doing. There's an application it doesn't recognize in our computer. And yet that app keeps talking out to the same URL over and over and over again and maybe sending some information out to URL. It would be nice to have an anti-spyware program identify that, show you that this odd activity is occurring, and make you wonder exactly what that might be. So that might give you a little bit of a heads up whenever you're looking for spyware on your computer and something might be there that normally is not seen by other applications.

If you have email, then you certainly have spam. It's a normal part of doing email on the internet is you're going to get people sending mail to you that you did not ask for. Sometimes this email that comes to you is for you to buy things. And they wouldn't be sending all of this email if somebody along the line didn't click on it.

It's so inexpensive to send an email. It costs practically nothing, so why not send out a billion email messages? And if 0.001% of those people actually click and buy something, it's worthwhile to the spammers. So they don't mind blasting all of these messages out there.

What you really have to watch for are these messages coming through that look like they're from somewhere legitimate. They look like they're from YouTube. It looks like it's from your bank. And it's asking you to click that information.

That's a phishing attempt to try to get you to give up a username and password. You click the link. It looks like you're going to your bank. It looks like you're going to YouTube. But the reality is it's a fake website that has nothing to do with your bank and nothing to do with YouTube.

Many of your email clients that you're using these days have the anti-spam capabilities built into them. If you're using a web-based service from Yahoo or you're from Google, it's already looking for spam. You have a separate folder that has already been set up for spam.

I use Thunderbird on one of my computers. And Thunderbird also has anti-spam technology built into it. Creates some spam folders, looks for that, so anything coming in gets automatically moved off to the spam folder itself.

If you're in a larger organization and all of your email is coming inbound, you may have a third party in the cloud scanning the details for you for the spam. So that before it even gets into your building, it's already been scanned by a third party. Some of those services can really, really help keep down the amount of spam you might get in your environment.

When JavaScript became popular in our browsers, the advertisers figured out that they could create new windows. They could have those windows a certain size. And they could

put whatever they wanted inside of those browser windows. And it would pop up right on the top of everything else.

And what advertiser doesn't want to get their message right there in front of you on your eyeballs? The problem is that pop-up messages are horribly annoying and very quickly that type of advertising fell out of favor. The bad guys, and in some cases the not so nice advertisers, are still trying to use pop up though.

So many of our browsers have an anti pop-up or a pop-up blocker built into the functionality of the browser. And that way you can turn it on or turn it off as you'd like. These can really take over the screen.

The malware developers have realized that they can pop up a message that says, we've identified a virus on your computer, and it pops right to the top. It makes you think your computer is the one that's identified this. And they get to download code that way. So pop-ups are generally not a great thing.

However, of course, some pop-ups might be completely legitimate. You might have a banking website that tracks your time online. And when you're going to time out, it's going to log you out automatically. And pops up a message, a pop-up window, that says it's going to do this in the next 60 seconds. And it will tell you when you're logging in, make sure you turn your pop-up blockers off.

And fortunately, our browsers are configured in a way that we could turn it off for our banking website, but leave the pop-up blocker on for everything else. That way the applications that need it can absolutely present pop-up messages to us. And we can block all of the advertising and malware from other sites automatically.

These days every operating system comes with its own host-based firewall. And generally, these firewalls are turned on by default. This is a software-based firewall. It's a personal firewall that runs on our computer that is going to protect us from other people that might be on the network. So your personal computer is now its own self-contained system that is checking every bit of network traffic that's going in or going out.

So this is really nice when you have a laptop. You're going to a coffee shop. You're going to a hotel. Those are usually very open wireless networks. There's not a lot of encryption included by default and information could be going back and forth. People you don't know in another hotel room may unintentionally or intentionally have access into your computer.

But with the host-based firewall there, you're preventing anybody from gaining access to your computer without you specifically allowing it. And if you're traveling a lot, that's definitely what you want. You can usually restrict this activity by port number— traditional firewalls work with port numbers, **TCP** and **UDP** based port numbers.

But the host-based firewalls know what applications you are running, because they're running right there on the same operating system. And that gives them a little bit more flexibility, because you could say, allow this FTP application to talk out, but don't allow anybody to connect to me using this FTP application as the service.

That gives you a lot of security. And by customizing these firewall rules that you might have in your system, you can really create a very, very secure system, no matter where you might go.

**Tags:** [certification](#), [comptia](#), [firewall](#), [malware](#), [pop-up blocker](#), [security](#), [spam](#), [spyware](#), [virus](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Patch Management – CompTIA Security+ SY0-401: 4.3**

Patching your operating system is a good way to stay ahead of the bad guys. In this video, you'll learn how operating systems are patched and why the patching process may not be as easy as it looks.

Managing patches on your operating system is very similar to managing the patches for applications. We spoke of that in an earlier video. And it's just as important for our apps. We want our operating systems to be secure. We also want our operating systems to be stable.

So there will constantly be updates that come out for our operating system to work. In fact for Windows we get at least one update every month. And we always have to be sure that we're keeping our systems up to date.

Sometimes we get service packs, though. After a certain amount of time goes by you've got a lot of different patches. And when you bring up a new operating system for the first time we have to make sure that all of those patches are installed.

But if it's been six months, and every week a new set of patches has come out, then we may end up having to download install a lot of patches. So one of the things that Microsoft does is create these things called service packs, where you can install all of the patch is at one time. And you can even integrate them into the operating system installation so that when you install the operating system for the first time it may be at a certain service patch or service pack level.

And that makes it much quicker. If we know we can install Windows 7 and automatically have it service pack one, we only need to install the patches that have occurred since service pack one was released. It makes a little bit faster for our patch management.

As we mentioned, these updates are usually going to occur every month. You're going to get incremental updates. And these are going to be relatively important updates.

Microsoft doesn't release an update unless it's something that does affect security and stability, and it puts it into different categories. It tells you about important updates. And it tells you about optional updates.

If you're updating a driver in your computer that does not have a security concern associated with it, maybe that's an optional update. But if this is part of the operating system, where a security problem has been found, it generally gets put into the important update category.

Which brings up another point, which is what if you find a really, really bad vulnerability before the update time comes at the every month time frame? Well then that is something called an out-of-band update, where Microsoft has been presented with a problem that affects a large number of end users and it's too long to wait to put this patch out. So they'll create an emergency update for the zero-day and these other important vulnerability so that your operating system remains as secure as possible.

In our previous video, we spoke about the type of updates. But I thought a review would be useful depending on how you manage updates in your environment.

For Windows, you may want each individual machine to do a Windows update, which is 1 by 1 by 1. Or if you're a larger, domain-type environment, you might want to take advantage of Windows Server update services so that you have one central server, and you get to decide how you roll those patches out. Apple, of course, and Linux, has also

other options to be able to update all of the operating system patches, security updates, and everything else for those operating systems.

This update process for operating systems isn't exactly seamless when you get into a large and complex environment. For our home computers, we tend to turn on updates and just have it download the updates and install them. We don't care very much.

But we also have to consider that, when we are installing these updates, they might introduce other problems. They might break an application that we use in our environment that is a mission critical app. This application must run for us to be able to perform the duties and functions of our organization. So you don't want to install a patch and suddenly your entire business comes to a complete halt.

So sometimes you have to pick and choose exactly what patch you want to install. And that's when that Windows update server can be really, really useful, because you can deploy the patches as you would like. Maybe you'd like some patches to be installed, other patches not to be installed. You can control that all from a central place.

That central management really gives you a lot of flexibility not just with the type of apps that are being deployed, but also the bandwidth, because that central update server is the only one downloading the patches. Everybody else is getting their patches from this internal server. And you're saving a little bit more bandwidth on your internet connection.

No matter what system you use, or how you deploy these, patch management becomes an incredibly important part of your overall security strategy.

**Tags:** certification, comptia, patch, security

**Category:** CompTIA Security+ SY0-401

### **White Listing and Black Listing Applications – CompTIA Security+ SY0-401: 4.3**

One common strategy of managing applications is to set specific restrictions on their use. In this video, you'll learn about the advantages and disadvantages of white listing and black listing applications.

One of the fundamentals of the security of your host is based around what applications are running on your computer. It's these running applications that are able to take advantage of buffer overflows or get some type of injection of data onto your system and infect you with malware or spyware or allow somebody access to your resources. Any application can be dangerous. And so a Trojan horse, a piece of malware, or anything else might be hidden inside of these applications.

Since these exploits can only happen if an executable is run, it makes sense for the security administrator to think about ways to control the execution of applications. And there's two major strategies here. One is to have a **whitelisting of applications**, and another one is to have a **blacklisting of applications**.

If you're going to **whitelist applications**, that means that nothing can run on a computer unless it is first approved. This is obviously very restrictive, and it probably requires quite a bit of administration, to first have a list of approved applications, and then certainly there will be applications that need to be constantly added to this list. The advantage, of course, is that you know exactly what applications are going to be running on these computers and nobody's going to be running applications that have not first been checked to make sure that they're safe for your system.

**Blacklisting** is the exact opposite. You can run any application on your computer except for these specifically named applications. We see this very often with antivirus software or anti-malware software, which has a set of signatures for certain applications, and it simply will not allow you to run those apps on your computer. This obviously requires a

lot less administrative overhead, but it also is opening up your system to run applications that perhaps have not been blacklisted yet. And therefore, you have to stay one step ahead of the bad guys.

When you're dealing with host security, the decision to run or not run an application is something that's usually built into the operating system itself. These decisions are usually made based on a blacklist or a whitelist, and they're configured, set up, and managed by the security administrator. If we're going to add an application to a whitelist or a blacklist, we'll need to identify it in some way. And identifying it just based on a file name is not something that's very secure, since other applications can simply use the same file name and get around that kind of restriction.

Instead, we tend to use something like an application hash. We get a hash value of the contents of that executable, which means that every executable's going to have a very unique hash identifier. Even if you have two file names that are identical but the executable is different, they will have different hash values.

So you'll be able to designate what applications are good or bad based on this very unique value. And even if the bad guys are able to modify the executable by embedding their own code, the hash will then be different. And therefore, the executable will not be able to run on that system.

Many application developers and publishers will digitally sign their executables. They use a certificate that everyone has so that everyone can automatically trust this digital signature. Microsoft does this a lot with their executables. So when you run Microsoft Word, that executable has already been digitally signed by Microsoft.

One good example of how to prevent this, obviously, would be to set up a policy that allows digitally signed executables for Microsoft but will restrict any other kind of executable on the system. An application management strategy that is less secure but probably easier to administer is configuring your system to run certain applications from a particular path on your computer. This means that only applications that are on certain folders inside of your computer would be able to execute.

And on Windows systems, you have this concept of a network zone. And if you are in a particular network zone and running a particular application from devices or servers in that network zone, it might be allowed. If you happen to be outside of the building and you're in an external or different network zone, perhaps you'll restrict applications from running in those particular unknown or untested zones.

**Tags:** application, black list, certification, comptia, security, white list

**Category:** CompTIA Security+ SY0-401

### Trusted Operating Systems – CompTIA Security+ SY0-401: 4.3

It takes a lot of work to certify a trusted OS. In this video, you'll learn about trusted operating systems and how much time and money it can take to validate this trust.

In working in very high security environments, you may hear the term trusted **OS**, trusted operating system. And this comes from the idea that our operating system will have been created, developed, designed, tested, and evaluated to be sure that we can trust what's happening inside of that operating system. And it's based on something called an **Evaluation Assurance Level**.

You'll hear this most often referred to as the **Common Criteria for Information Technology Security Evaluation**. There's a lot of words there, so most people just call it **Common Criteria**. You may see abbreviated as **CC**.

This is an international standard. So this is one that is a very well known. And you often see it related to government type work. Especially the US Federal Government, and perhaps other governments around the world, take advantage of this because it is a very common and universal requirement and set of standards. It's one that many, many different manufacturers can write their products, their hardware, their firewalls, their security products to meet these common criteria requirements.

When something is tested with these common criteria requirements it's given an **Evaluation Assurance Level**. The higher the **Evaluation Assurance Level** then the more testing and the more evaluation, and ideally, the more secure a product might be. And you'll see these referred to as an **EAL1** through an **EAL7**.

And where we're talking about operating systems, and how they work, and how they're developed, and how they're tested, when we talk about a trusted operating system we're usually referring to one that has any type of **EAL compliant level**. But the most generally accepted one for a trusted **OS** is that it's at a minimum of an **EAL4**,

To get an idea of what manufacturers of these operating systems and security devices are going through to get their devices at an EAL4 level, I grabbed these stats. This is from the United States Government Accountability Office. This document is **GAO-06-392**. There's the URL if you want to download the PDF. And it shows just how long it takes to get something to be EAL compliant, and something that has been tested and signed off as being EAL4 compliant not only from a time perspective but a dollars perspective.

You can see something for EAL2 may take anywhere from five to just under 10 months to get that certification. EAL4 goes from 10 months perhaps all the way up to 24 or 25 months. It could take years to get that particular device, software, operating system, to be EAL4 certified.

And it doesn't come cheap. For EAL2 you're spending anywhere from about \$75,000 up to \$200,000. For EAL4 it's \$150,000 up to \$350,000 to get that certification completed.

Obviously, the manufacturers that are putting their devices through this type of certification, that are spending the time and spending the money, are doing it because the federal government needs very, very secure systems. And that's why you not only see the government using these EAL certifications, you also see private organizations using them as well. Because they'll look at the testing the government did and say, well if they spent all of that time and all of that money evaluating it at that certain level, we can also be sure that the operating systems that we're going to use are trusted operating systems.

### **Host-based Security – CompTIA Security+ SY0-401: 4.3**

On a public Wi-Fi network, all of your security is whatever you've configured in your operating system. In this video, you'll learn about host-based firewalls and intrusion protection systems.

Host-based firewalls are an excellent way to protect your system from the bad guys coming inbound to your computer. They set up a wall between you and the outside world so that people don't have unfettered access to everything that's inside of your computer. You often hear these referred to as personal firewalls. And you'll find them in many different operating systems. Certainly if you're running Windows, or Linux, or OS 10, there is a personal firewall or a host based firewall that's already installed, and probably running in those operating systems.

You can also find third party solutions available. Many anti-virus and anti-malware companies will also include their own firewall along with their anti-malware software. So whether you use the built in firewall in your operating system or you use one from a third party, they're all designed to keep the bad guys out of your system. These host based firewalls are stateful, which means they keep track of all of the sessions that you're creating to the outside world. So you can continue to surf the internet without any type of interruption.

But just because you're surfing out doesn't mean the bad guys are able to then come into your network. It's the stateful firewall that keeps track of this and allows your communication, but prevents any communication that doesn't fall into that state. These personal firewalls are also able to manage the communication by application. This is something that's very difficult to do on the network side. But if you were embedded into the operating system itself you have the luxury of knowing exactly what applications happen to be running on your system, and you can allow or disallow access in and out of that application all from this built in firewall.

The Windows Firewall is a good example of a personal firewall that's able to use this application visibility to be able to make decisions about what traffic goes in and out of your system. The Windows Firewall can also be configured to allow or disallow traffic based on a TCP or UDP port number, which means that it can span many different kinds of applications, and really have a much broader security policy that's based just on a single port number. Here are the configuration settings for a Windows Firewall.

And whether you're using Microsoft Windows or any other operating system, the configuration settings are very similar across operating systems themselves. The first dialogue that we have here shows the ability to turn on and off the Windows Firewall. This is where you might want to turn it off, which is, as you can see, not a recommended setting. Generally, you want to have your firewall on all the time. In fact, if you do go outside of your house, maybe you're got your laptop with you, and you're going to a open access point, you're going to a coffee shop, you may even want to turn on the firewall and include the option to block all incoming connections.

You may allow incoming connections inside of your home. But when you're outside of your home, that is an even safer configuration to prevent anybody from using any of the exceptions that you may have created in your firewall. And creating exceptions is a very useful tool. You can specify a particular application name for instance, and you would allow access into your computer using that particular application. For instance, Remote Desktop.

I might configure this system to be accessible from outside using remote desktop. No other application would be able to work unless I also included exceptions for those individual applications. And if you would like to have a much broader security profile, you could even add a specific TCP or UDP port number. That would allow access from the

outside. Regardless of what application happens to be, it would just need to be able to access your system using these very particular TCP or UDP ports. Just as we have network based firewalls and network based intrusion prevention systems, we also have those that are designed for individual hosts.

And you're starting to see more and more individual host intrusion prevention systems running in software. These are usually separate applications. They are sometimes integrated into the firewall, especially if you're using an IPS on your host that is built by a third party that likes to integrate all of these applications together. These host-based intrusion prevention systems generally protector system based on a number of signatures.

So it's going to look for a certain number of things to occur on your system. And if that traffic happens to match this very specific signature, it will alert you to this, or it will block that particular function altogether. So it can also look for certain types of activity. It may not know the signature for modifying a particular file in your system, but it knows that no one should be going into your system 32 directory and modifying anything that's inside there.

And if it does, then it will notify you that I don't have a signature. But this looks awfully suspicious. I wanted to make you aware. Or it may choose to block it completely. So you may want to make sure that your system is running your built in firewall, and perhaps even has an intrusion prevention system to make sure that you're able to maintain the security of your computer.

**Tags:** [certification](#), [comptia](#), [firewall](#), [host-based](#), [intrusion protection](#), [ips](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Hardware Security – CompTIA Security+ SY0-401: 4.3**

Our physical devices need as much security as our operating systems. In this video, you'll learn how cable locks, safes, and locking cabinets can help keep our systems under our control.

We not only have to think about securing the software and our networks in our environment— these are very virtual things. They're bits and bytes. And they aren't something we can physically touch— we also have to protect our hardware, because our hardware systems— our laptops, our mobile devices— these devices are ones that people can take. They can damage. They can create problems for us.

One way to protect some of these more mobile devices, like our laptops, is to get something like a cable lock. This is really temporary security. You might go in the morning and lock your laptop to something solid, lock it to the desk that you're using, lock it to the leg of the table. So it would be very, very difficult to lift up that table to try to get that laptop out of there. This all works really almost anywhere— if you're in an airport, if you're in a hotel, if you're in a conference room— that way you can leave the laptop there and at least be relatively sure that's not going to be easy for that laptop to walk away.

Most devices— if you look at the side of your laptops and your mobile devices— there's a little notch there— it's a reinforced notch, usually reinforced with metal— that you plug this particular lock in. You turn the key. And it's now locked in there. It's not coming out. And it's a really, really easy way and a very quick way to lock your laptop or mobile device right to a box, to a system, to a table. It's not going to go anywhere from there.

Obviously, this is not long-term protection. You can see this little cable is very, very thin. They have thicker ones.

But even so, a nice pair of cutters will come in, cut that cable very, very quickly. Some people have become very good at picking these round locks in different ways. There's plenty of YouTube videos out there that can show you some of those techniques. So this is something where you're really protecting your laptop to a certain point, but don't rely on a cable lock to provide long-term security for your laptop. You can't leave your laptop in one place and expect overnight for it to be there just because you put a cable lock in place.

A more permanent security technique would be to put a safe in place. And this is also a good way to protect your backups and your other media from anybody else gaining access to it, especially when we talk later about encryption technologies and being able to have a certain key available that we decrypt all of your data. You may want to keep that key, that software encryption key, inside a safe in your environment.

So you can also protect your laptops and your hard drives. You get very large safes. You can get smaller safes. And generally, you would get a safe that has a little bit of protection against the elements. They have these fire safe so that if your safe is in a building that has a fire, and it doesn't get too hot where the safe might be, it's designed to protect it up to a certain amount of heat.

Also water is a concern, especially in flood areas. You might want to get a safe that is airtight, that would not allow water to leak in to the media that you might have in there. Obviously, these safes are very, very big and very, very heavy, so they become difficult to steal. Our laptops walk away so easily because they're so light and so small. But if we've got the laptop inside this big safe, we can be assured that nobody's going to be easily walking away with something so big and so heavy.

We also have to be careful about managing the safe combination. We have to think about who would have access to the combination. We have to trust that that combination would not get out to other people.

And then we also have to think about what would happen if we lose the combination. Is the safe one that we can drill into easily? It might damage the safe. And it might take time. But at least we would have access to our media. All of these things combined really create a very, very secure environment if you need to lock something up and protect it over a long period.

If you've ever been into any reasonably sized data center, there are a lot of different pieces of equipment in there. And they're usually managed by different people in the organization. You might have servers that are managed by the server team. You might have firewalls that are managed by the security team. There's other types of equipment for the phone systems that there may be a completely different telecommunications department that manages all of that responsibility is going to lie with the owner.

And because data centers are generally so open, you might want to consider getting a locking cabinet, so that only your department might have access into this. And if there was anything that was to happen to that hardware, you know that it could only happen by somebody who has that key. You can take these racks, install them side by side, and have multiples there. They all might open with the same key or with different key.

Usually, you have ventilation in these racks in the front and the back. You can see some of these ventilation slots that are here. And, of course, the top and the bottom, there's ventilation there as well. Sometimes there's fans. So even though your locking it up, you still can keep air flowing through and keeping all of those systems cool.

By using some of these hardware techniques, we can then be assured that our laptops, our media, and anything else that we have in our environment is going to stay as safe as possible.

### **Host Software Baseline – CompTIA Security+ SY0-401: 4.3**

How safe is your software? In this video, you'll learn how to create a security baseline on a server and considerations for baselining cloud-based devices.

To be able to secure applications, we need to understand everything we can about how the application operates. We need to know what the application is communicating with across the network, what type of traffic is being sent back and forth. We need to understand what's inside of those packets being sent over the network. We also need to understand what resources are going to be needed on individual hosts, what type of network connectivity is going to be required, and what should be expected whenever this application is running.

We also need to tighten down the operating system itself. We need to install and configure our host-based firewall and make sure that it's only going to allow this particular application to work properly. That way, if somebody's trying to send traffic into this application that does not look quite right, we've got a firewall to be able to restrict some of that communication.

We can also set up application execution restrictions, which means that certain applications may not be able to run on that particular host. That way, if the bad guy does manage to infiltrate that system, he would not be able to execute code that would be used to then exploit that particular app.

And you might also want to limit what specific folders this application has access to. Some applications might be told to write information to a system file in your operating system, which would certainly be outside of the scope of most applications. But if we can limit this application so that it only has access to write in certain areas of the operating system, we can avoid any of those types of exploits.

Once you've gathered this application baseline and you understand how the application communicates, you can then use that information to configure external devices like firewalls and intrusion prevention systems. Those can be locked down to only allow the proper communication. And if anything goes out of the scope of what's normally seen for this application, you can choose to disallow that communication through the network.

It's unusual these days to find an application that's centralized in a single area or on a single server. These days, the applications are very decentralized. They can be located in many different areas. Part of the application might be in a data center in one part of the world, and the other part of the application may be in a completely different part of the world. So we need to think about how we're going to secure the application, even across these very large areas.

We might also have other services and other applications running on the same physical hardware. We certainly see this with virtualization technologies these days. So it is a little bit more difficult to configure security settings and security profiles to a single device when you might have one, two, or even a hundred different applications all running on that single piece of hardware.

You may also require some additional redundancy. Part of security is making sure that your applications are available. So you want to perhaps have multiple locations where this application can run. If you happen to lose a facility, the other facility might still be running. This might also help you in cases where there might be denial of service attacks.

There could be hardware failures or power outages. Or you might have network problems. And by decentralizing the application itself, you may be able to make it much more resilient and avoid these types of security concerns.

### **Virtualization Security – CompTIA Security+ SY0-401: 4.3**

Our virtual systems have changed the way that we think about security. In this video, you'll learn about virtualization technologies and some of the more important security considerations in a virtual environment.

**Virtualization** is the concept where you can have one physical device. And on that one physical computer you can run many different operating systems or many instances of many different operating systems. These would all be separate independent operating systems that have their own CPUs, their own memory, their own network configurations. But they're all really running on one single, physical device. On our desktops, we can have host based virtualization where you can be running a **Windows desktop**, a **Linux desktop**, a **Mac OS 10 desktop**.

But then you might also have virtual systems running on your desktop so that you might be running a Windows or a Linux operating system in separate windows at the same time on your computer. There's also more of an enterprise level virtualization where you have a very large server that has a lot of memory, a lot of hard drive space, and a lot of CPU's associated with it. And you can spin up many, many different servers, sometimes hundreds of different servers all running on this one physical platform. This idea of virtualization's been around for a very long time.

The mainframes that **IBM** created in 1967 were using virtual systems on their mainframes. We've simply taken that idea through the years and honed it to run in **RPC** architectures. Here's a screen that gives you a good idea of what host based virtualization would look like. This is a **Mac OS 10 desktop**. This is my desktop. And on my Mac OS 10 desktop I have running a window for my browser.

So this is running in the native Mac OS. I also have a Windows system running. And this Windows is a self-contained unit. It happens to be running in a window here. But I could also make it run full size. So my screen looks just like I'm running on a Windows device. And in fact, you really are running Windows inside of this virtual system. And I can run any application I'd like. I've also got Linux running at the same time on my system. So this is how virtualization can really take the idea of using multiple physical devices and combine them all together so that you're running all of these operating systems on one single, physical device.

To bridge this gap between the physical world and the virtual world you need some specialized software called the hypervisor. This might also be referred to as the virtual machine manager. And it's in charge of keeping track of all of these CPUs that are in use and the memory usage, and making sure that the virtual platforms are able to use the proper resources that they're gathering from the physical world. Many of these host based systems will require a particular kind of CPU in your computer that supports virtualization.

There's specialized hardware in these CPUs that will allow a much more effective way of virtualizing your hardware to use across all of these different operating systems. If your CPU does not have this virtualization capability in the hardware it may still allow you to run the virtualization software. But the performance is not going to be as good as if you have that specialized virtualization capability in the CPU of your hardware. This hypervisor is going to be responsible for sharing the resources between the physical and the virtual systems.

So it will manage the sharing of CPU and memory. It'll make sure the networking pieces are all separated out. And it will make sure that there is security between all of these separate operating systems. Having the ability to have all of these different virtual systems and different operating systems running all at the same time certainly provides you with additional functionality of your computer. But it also provides you with some very nice security features. Each one of those virtual worlds is called a guest.

And each one of those guests has its own virtual file where everything is self-contained. If you wanted to grab that single file, pick it up and move it to another system, you could run that virtual system now on the new computer. It makes it very portable, and it also makes it very secure because everything is self-contained in that single file. Since it is a single file, you could do versioning of this system. So you might take snapshots occasionally of your operating system.

And if you happen to install a file or you get infected with malware, you can simply roll back to a previous snapshot. And your system now is in the same form that it was when you took that snapshot originally. You can store multiple snapshots. So every time you want to make a major change to your system, you simply take a snapshot. And it's very easy to roll back to a previous version. If you did also want to roll back to a particular date and time, especially if you're making a very big change to the operating system— you might be upgrading to a new service pack, or you might be installing new hardware into that virtual world, but that patch broke something associated with the update— this is very easy now to roll back to a previous version using the snapshot function.

Another nice security feature of these snapshots is that you have a way to go back in time to see when something may have changed. If the bad guy took advantage of a vulnerability and exploited your system, you could then go back to previous snapshots to see exactly when that occurred. And it may give you a little bit more of a determination of from a calendaring or time perspective of the date and time when you first saw this particular vulnerability become exploited. In organizations that are using this virtualization capability in their data center, they have a number of additional functions for performance.

One of these is called elasticity, which allows an organization to quickly make more systems and roll out more capacity when they need it, and then pull that capacity back, and run fewer systems when they don't need those capabilities. If there's a certain time of the day, or a certain period of the quarter, or maybe during the holiday season where you would need more computing resources, it's very easy to simply click on a particular image and deploy more and more systems out to cover that particular excess load that's coming in.

This also can be orchestrated. This means that you're able to set up some automated processes so that every time you deploy a new system, you click that button it not only deploys the new server, it loads the proper software for that server and then changes firewall rules to allow access to that new server. That's just one example of how this orchestration all works behind the scenes so that you can easily deploy a server along with all of the support systems that go around it.

You've also got the ability to do this across data centers. If one data center very busy but you have some excess capacity in another data center, the orchestration software allows you to simply drag a server off to another data center. You're now using those available resources, and your orchestration ensures that all of those support services will work properly once that's now running in the new data center. You can take advantage of some of these virtualization features in your security work.

If you need a new machine you don't have to purchase new hardware. You simply spin up a new virtual system running on your desktop or running in your virtual environment. This is a very fast way to spin up a system to use for port scanning or vulnerability testing. You can perform all the testing that you need. And when you're done, you simply remove that machine and allocate those resources to other virtual systems in your environment. You might also use virtual systems for testing software.

You can create a sandbox. And before you run software on a machine that's out in your production network, you might want to spin up a virtual machine, run that software, and see if there's anything detrimental that happens when you run the software in the virtual environment. And once you become comfortable with the software running in the virtual

world, you could then deploy to other systems in your environment ensuring that you've already tested and confirmed that that software is not something malicious that you should be concerned about. You might even find specialized software that runs on every single person's computer that acts as a virtual sandbox.

And as they are running executables for the first time, it runs it in the virtual sandbox on that particular computer. That's obviously a more advanced function of virtualization. But it's one that has given us more capabilities in the security world to keep all of our systems safe.

**Tags:** certification, comptia, elasticity, sandbox, security, snapshot, virtualization

**Category:** CompTIA Security+ SY0-401

### **Cloud and SAN Storage Data Security – CompTIA Security+ SY0-401: 4.4**

Our data is becoming increasingly distributed, and our security strategies have changed to keep up with these dramatic changes. In this video, you'll learn some best practices for protecting the data that week in the cloud and in our **SAN**.

The data controls that we apply to what we put into The **Cloud** are very similar to the same controls that we might use if all of the data was inside of our building. The difference, of course, is that in The Cloud there's much more of a security concern, since now that data could possibly be accessed by anyone wherever they happen to be. So you of course need to apply things like security controls to that data. You want to make sure that the proper rights and permissions are applied not only by the applications that are accessing that information, but also by all the administrators that have to maintain and keep all of that data working out in The Cloud.

We should also think about how we're storing this data. We commonly refer to this as "**data at rest**." It is on a storage device. In this case, it's in a storage device that's in The Cloud. We have to think about encryption of this information to help keep it private. If we store this information in an encrypted form, then obviously if somebody did gain access to that data, they wouldn't be able to read it. But of course, there's computing overhead and additional administrative concerns whenever you start encrypting all of the data that you're storing somewhere.

And of course, if all of this information is stored off site somewhere, we might want to add some additional security controls to it. We might want to put an additional firewall in front of all of this data, have different security access controls to that data, have intrusion prevention systems that can look for somebody trying to take advantage of vulnerabilities in those services that we're providing. Since all of that data is somewhere outside of your direct control, it's therefore at risk and you need to apply the appropriate security controls to make sure nobody comes across or has access to this information.

As the saying goes, the network is the computer. And since we also have all of this stored somewhere on the network, we could also say the network is the SAN. That **Storage Area Network** is gathering and keeping all of our information there for us and it's accessible from wherever we happen to be. We do want to make sure that we physically secure the Storage Area Network. If somebody does gain physical access to our storage devices, they can not only grab information from those devices, they can obviously damage or in some way make those devices unavailable to us.

So we need to make sure they're behind locked doors in a data center or some physically restrictive area— one that we can check and see who's going in and who's going out of that area. If you've ever been to a data center, there's at a minimum a lock on the door and there's generally additional security controls added to that particular lock.

Sometimes we might want to even consider having drives that will encrypt the data in hardware as it's written to the drive and decrypt the information as it's coming off of the drive. That way, if somebody did break in and they stole the drive itself, they would not have access to that information without the proper security pass phrases or the security certificates.

We also might want to consider how this information is stored if it leaves this protected area. Good example to this is when you're transferring data outside of the data center—maybe to somewhere else in the organization across a wide area network or to a third party. You may want to encrypt that data as it's going across the network.

One additional concern with transferring data or having data as it leaves the data center is with your backup tapes. These backup tapes are very often stored at a third party facility. Unfortunately, backup tapes can go missing as they are transferred between locations. So you want to be sure that if a tape is missing, that at least nobody would be able to gain any information from that tape because you've encrypted everything written to that tape.

We also have to think about how encrypting data may have an effect on the resources that we're using. When you encrypt data, it doesn't come for free. There's additional CPU cycles. It may require additional traffic to go across the network. There's some type of overhead of resources. A backup that took an hour may now take longer than an hour and is that something that we can tolerate given the amount of time that we have for backups?

All these things have to be considered when you're taking this data outside of that trusted environment. And sometimes it even has to be considered if you're keeping it in that trusted environment just so you can be sure that everything that we're storing on these devices will be protected.

The concept and implementation of big data is completely changing how we think about the protection of data. This big data is obviously a huge massive dataset. We're taking information from many diverse data sources and we're storing it in one central place. That means that the normal access controls that we might usually apply to a certain type of known data may not apply to this big storage of data that we have.

You can usually fit a "need to know" principle to a traditional dataset. If you're in the accounting department, you can access accounting information. If you are in the marketing department, you can access marketing information. But with big data, we may not even be completely sure exactly what type of data we're storing.

That's one of the objectives of big data is to store everything you have and later on, we will sift through the data to try to find relationships and build some intelligence from the data that we've got there. That's the idea of hunting down patterns. And we don't know what patterns we're going to find until we go hunting for them. So now becomes difficult to qualify who gets access to the data and who doesn't.

As we're taking this information from all these very diverse data sources and we're pulling it back into this big data repository, we may want to consider filtering out personally identifiable information or PII. This might be a social security number. It might be a telephone number. It may be anything that might be personal data that we might be concerned with somebody else getting their hands on. And if we can filter that out before putting it into the database, then we can perhaps relax our security controls because we know if somebody was to get access to that big data, they still would not have access to any of your personal details.

Once we have all of this data stored in this location, we may then want to think about what information people are pulling out of the data. The problem is that it becomes very difficult

to audit this because a big data repository may have many different queries going on. And there may be a massive amount of data coming out of that data store.

So in those particular cases, it might make more sense to simply store the queries that are made to the database. Later on, if you wanted to perform an audit and see exactly what somebody was retrieving from the database, you could then simply perform exactly the same query and then see the exact response that person received.

This might also prompt you to implement some **DLP** into your environment— or **Data Loss Prevention**. A **DLP device** sits on the network and it watches for information to flow over the network. And if it notices information that should not be transferred— like social security numbers or credit card numbers or health care information— it can filter out, limit, or alert when those particular pieces of information are transferred across the network.

By utilizing some of these data security strategies for your Cloud data, your SAN, and your big data, you can be assured that the information that you're storing and the information that you're querying is going to be as secure as possible.

**Tags:** [certification](#), [cloud](#), [comptia](#), [san](#), [security](#), [storage](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Data Encryption – CompTIA Security+ SY0-401: 4.4**

If you want to keep your data safe, then you need to encrypt it. In this video, you'll learn about encrypting full-disks, databases, individual files, removable media, and data on mobile devices.

If you work in an environment where the data on your device is so important that you want it to be protected any time you're away from that computer, you may want to consider full-disk encryption. This is an encryption methodology that encrypts everything on the drive. It encrypts all of your documents. It encrypts all of the files. The operating system itself is encrypted. Nothing is left unturned. Everything on that drive is now protected with this encryption mechanism.

This is obviously perfect for mobile devices— devices that can easily be stolen, devices that may get out of your hands and into somebody else's. And if you have the entire disk encrypted, you can feel very secure that even if somebody does gain access to the hardware, they're still not going to gain access to any of your data.

Many of these encryption methods are built right into the operating system. Linux has a number of encryption options available. In Windows, you could use **BitLocker**. In **OS X**, you could use **FileVault**. And it makes it very easy because it's part of the operating system itself. Of course, you can go outside of the operating system and get your encryption technology from a third party. For example, **PGP** has a full-disk encryption application. And up until recently, **TrueCrypt** was a very popular full-disk encryption option for Windows, and that project is now inactive.

With any of these encryption methods, the management of the keys is incredibly important. It is these certificates that are encrypting this data. So we want to be sure that we have a backup of that data. If that encryption key goes missing, or you lose it, or it's damaged in any way, you will no longer have access to any of the data on the disk. So it's always important if you're using full-disk encryption, that you've also got a very good backup or you have at least copies of your key that you're keeping other places.

If we're keeping a large active store of data in a database, it might make sense to us to simply encrypt all of that data. But technologically speaking, it becomes very difficult to have all of that information stored in an encrypted form and still have a very good way to

access that data in a timely manner. Encrypting data requires CPU overhead. There are calculations that have to be made. And if we're retrieving and storing information very quickly— especially across a very large number of people to a very active database— it may not be practical to keep all of that information encrypted.

The ability to encrypt this data at the database level may be subject to the capabilities of the **DBMS** that you're using, that is, the **database management system**. So there are some methods within **Microsoft SQL Server**. There are completely different encryption methods in **MySQL**. Those engines are very different, and as you go across different platforms, you'll see that there are different encryption options, depending on what you're using.

What many people will do is instead of encrypting the entire database, they'll pick particular fields— especially fields that have very sensitive information— and will only encrypt that data. This way you're able to retrieve and store information reasonably quickly, but still protect information that would be sensitive inside of that database. It's very important if you do have a relational database, not to encrypt your key fields. These are the fields that will allow you to compare information when you're comparing across individual databases. And it's these indexes and these key fields that are very important to maintain in an unencrypted form.

If you don't want to encrypt everything that's on your drive— you don't want to take advantage of full-disk encryption— but there are certain files that would be nice to store in an encrypted form on your drive, you might want to take advantage of some built-in capabilities of your operating system. Many operating systems allow you to right-mouse click on a file, choose to encrypt it, and that file would then stay encrypted on disk. That way, if somebody did come across your computer, they may be able to see some other spreadsheets and documents on your drive, but the ones that you've encrypted would obviously be protected.

There's a number of third-party applications that can also do this, so you don't have to rely on the operating system. You can rely on the encryption method that makes sense for you. This means of course, that some files on your drive will be encrypted, and other files will not. And you have to decide, then, exactly what data will be stored in an encrypted form on your drive.

Remember that encryption doesn't come for free. There are CPU cycles that must be used to retrieve and store that data. There's memory that has to be used to be able to perform that encryption process. For example, if you have a file on disk that is being accessed quite a bit, you might be losing response time and speed by encrypting that data. And then it's up to you to decide if the decrease in performance will be made up by the security that's going to be based on that encryption.

In all of these cases, again, you want to be sure that you protect the encryption keys that are used on this information. Even though it's a single file, there's still going to be an encryption key. And if you lose that encryption key, you will lose access to that encrypted data. So make sure you have backups, make sure that information is stored elsewhere, and make sure that that key is something that you can then provide back onto the computer if it ever gets lost or damaged.

The security of the data stored on removable devices is the bane of the security administrator. And I can tell you personally that I've lost a number of USB keys through the years, and of course, wherever that USB key ends up, is where my data is now going to live. And it's very important that the data, therefore, is protected on those removable devices. The security administrators and system administrators in your environment may set policies in the operating system that require that data that is stored on a removable device must be encrypted. And they can automate this process, so whenever you store data on that removable device, it will always be encrypted without any input from you.

And this way, they can be assured that if that information is lost in some way, at least the data itself will not be accessible.

Again, key management is incredibly important. There is an encryption key, and that encryption key is what you use to retrieve that data. If you lose access to that key or that key is damaged, all of that information is no longer going to be accessible. It's very common for network administrators and operating system administrators to automatically store your encryption keys in a central area. That way if you leave the organization, or your laptop goes missing, or you lose the key, they can then provide a key that's going to gain access to that data.

And if your system administrator is very concerned about data on these removable devices, they might set policies that say that the USB is completely disabled on your systems. And in that way, you would not be able to plug in a USB drive or external hard drive and store any information on a removable device.

We're all walking around with our mobile phones and our tablets, and these devices obviously have data stored on them that are also a concern of ours. We want to be sure that we're encrypting that data, as well. Fortunately almost all the operating systems that you'll run into on these mobile devices already implement some form of encryption of the data on that device. It may not be all of the data on that mobile device, but some of the most important and private information is encrypted by default.

Very often, it's the key on this device that, of course, is going to encrypt all of that data. In fact, if somebody performs a wipe of data on these mobile devices, what they're really doing is deleting the key. Once the key is deleted, none of the data on your device will then be accessible. If you're using an iOS device like an iPhone or an iPad, then a lot of this data may already be encrypted, using something Apple calls Data Protection.

If you look into the configuration of your device and you've enabled a passcode, then you'll have a note there that says Data Protection is enabled. This means that if your mobile device is stolen, they would have to have that pass code. If they don't have the pass code, then they do not have access to the data. Not everything on your iOS device will be encrypted using this method. Things like SMS messages or pictures are not generally encrypted. So you can't be assured that everything on the mobile device will be encrypted, just some of the most important information on the device.

If you're running the Android operating system on a mobile device, there are encryption settings in the Settings and Security section. You even have the option for a full-disk encryption across that entire device. And the key, again, on these devices is built on the pass code itself. Regardless of the type of data you're using, whether it's on your desktop, whether it's on removable storage, or on your mobile devices, encryption becomes extremely important and a very valuable way to protect your personal data.

**Tags:** certification, comptia, data, database, encryption, file, full disk, mobile, removable, security, usb

**Category:** CompTIA Security+ SY0-401

## Hardware-based Encryption – CompTIA Security+ SY0-401: 4.4

You can add additional security features by using hardware to assist with the encryption process. In this video, you'll learn about trusted platform modules, hardware security modules, usb encryption, and hardware-assisted hard drive encryption.

If you hear the term “**trusted platform module**” or **TPM**, it's referring to probably one of two different things. One of those things is the standard that is used for cryptographic functions that's then applied usually onto a piece of hardware. So sometimes you'll hear of the TPM standard or specifications.

You'll also hear of TPM chips that are on the motherboards or our computer systems themselves. And that is where we'll have this hardware built into our computer to help us out with all of this encryption and these other types of cryptography functions that we run into these days.

One of the things that's on a **TPM** is a cryptographic processor. This is a processor that is built as a random number generator. It has key generators built in it. A lot of the heavy **CPU** usage that is done with doing any type of cryptography uses some of these standard processes. So having a piece of hardware that can do that rather than the main CPU of your computer is going to be helpful.

There's also inside of the TPM something called persistent memory. There are a number of unique keys that are burned into the hardware when this is produced. And those keys obviously can't be changed because they're burned into the hardware. That's really useful if we need to now have a key that's already pre-generated that we can then create other types of encryption methods with.

Another function of a TPM is versatile memory. This is memory where we can store information. For instance, we can store keys in our TPM. Another thing that's useful is you could have a piece of software scan a piece of hardware, scan the hard drive, scan the motherboard, scan the memory inside of your computer, and then store that information inside— cryptographically store it— inside the TPM.

And then when your software starts up again, it can perform the same checks, compare it to what it saw last time, and see if somebody might have changed out the hard drive. See if somebody changed the amount of memory inside of a system. And you know that the change occurred because you were able to cryptographically sign it and store it in a way that was secure. And you know if anything was to happen to that computer.

This information is being stored on the TPM and accessed via a password, so one of the things built into this very smart processor is a way to prevent brute force attacks or dictionary attacks of the TPM itself. That way you can be sure that your password isn't one that somebody's going to be able to find just by going through a huge list of dictionary words.

When you get into large scale or high end cryptography, you'll often run into these devices. These are **hardware security modules or HSMs**. You'll usually see them as **plug-in cards** or **PCI-type adapters** in a computer. They may also be a separate standalone hardware device. And they can do a lot of things for us. They can back up our keys and keep them in a very secure environment so that nobody can access those keys except for us.

These may also have on them cryptographic accelerators. So a lot of the things that our systems are doing to be able to encrypt and decrypt and create keys and validate keys can be offloaded onto one of these specially designed pieces of hardware.

You'll usually see these HSMs used in very large environments, especially ones where there is a lot of cryptography in use— financial organizations, credit card information, that type of thing. And usually you can cluster them together. There's redundant power that you can get for them. That way, they can be very, very redundant and reliable. And even if you lose one of them, you can be assured that your cryptography functions will still continue to operate.

Our mobile **USB** data drives now are also getting very, very smart. There's a lot of data encryption, hardware encryption built right into some of these **USB keys**. This is **hardware-based encryption** that's built as part of the USB key itself. If you have a key, you can be assured that the data on the key is always going to be encrypted. And we mean high speed encryption. It's **AES 256-bit**. It's very strong encryption that is on these USB drives.

There's security software that's also built into this. In fact, many of these come with a browser already as part of the USB key. Because you know that browser's one that's going to be trusted and you can browse the net and be assured that nobody can look in on what you're doing.

This can also be used as a secure token. So if you're carrying around a pseudo random number generator or you'd like to have a two-factor or method to single sign on, you can use these USB encryption keys to be able to perform that function as well.

And because these are so important and they carry such important data, they usually have a remote management function built into it. So if you happen to lose the USB key, an administrator can assign that key to be deleted the next time it has ever seen anywhere. Somebody plugs this into a computer, that USB key is going to talk out to the internet, it's going to realize that it's no longer in the hands of the person who originally had it, and it's automatically going to wipe everything that's on that USB drive. So some very nice remote administrator functions there as well.

We talked in an earlier video about full disk encryption. But that is software that comes as part of your operating system or software you can get from a third party to load on to your system. And it requires a little bit of software.

But there is also hard drive encryption that you can get that is completely invisible to the operating system. It is hardware itself where you would plug in a drive into this encryption device and then continue that back off to the motherboard. It sits in the middle between your motherboard and the drive itself. And so everything flowing through here gets encrypted. It is one that can also integrate with the USB key.

So you can step up to your computer, plug in your USB key, and only if the key is on that USB drive are you going to have access now to the data that's contained on that encrypted hard drive. It is something that is very well engineered where you can simply have one connection going in and one connection going out. It's very high speed. You don't even know it's there. And the encryption on here is very strong. So if somebody was to get a hold of your hard drives, they would still not have access to that data.

You can even chain them together. You might have a requirement that two people have to be there to gain access to that data. Both people have to plug their USB keys in and only then is the data available and on that hard drive. So a lot of nice hardware features there.

These days, with all of the types of data that we have and all of the importance associated with that data, we need to make sure that we're able to use some of these hardware-based encryption methods to keep that data safe.

## States of Data – CompTIA Security+ SY0-401: 4.4

The state of a piece of data will assist in determining the best way to secure it. In this video, you'll learn about securing data in-transit, at-rest, and in-use.

Of the three states of data that we're going to discuss, the first one will be data in transit. This is sometimes called data in motion because it refers to the data that's being transferred across the network. As the name implies, we're sending information across the network through switches, through routers, across wide area networks to many different devices on the network. And it's very important that we are able to secure that data as it's passing through. Obviously, not all of the data inside of this information can be encrypted because we still have to have headers and information that will tell the routers and the switches where to send the data.

And if we encrypt it, obviously the data cannot be read, and it will never make it to its endpoint. Instead, what we'll do is encrypt the data that's being stored within these headers, and within those packets. And we do this using a number of different methodologies. If you're communicating to a web server, you'll probably use **SSL** to communicate to that web server, the modern name for that being **TLS**, or **transport layer security**. If this is something that's being transferred across a wide area network or you're transferring it using a virtual private network, you're probably using a method called **IPsec**, which stands for internet protocol security.

Once we transfer the information across the network, we're probably going to store it somewhere. And we refer to this stored data as data at rest. It's on a hard drive, it's on a sand, it's on an SSD. It's on some device that's stored and waiting for us to retrieve it. Since the data is now being stored, it's probably a good idea to think about encrypting that data as a way to protect it. We can do whole disk encryption, which is very common to see on mobile devices or devices that you're very concerned about what happens to that data if it gets out of your control.

You might also want to encrypt parts of a database. If you're storing private or sensitive information, perhaps that's a perfect place to begin storing it in an encrypted form. And of course, you could maybe choose to only encrypt a single file or a group of files in a folder, and simply have that section of the data at rest be in an encrypted form. Regardless of whether that data is going to be encrypted or not, we still have to apply the appropriate permissions to that data. If this accounting information, then the accounting department may have access to look and change that data.

But you don't want shipping and receiving to have any access to that information. So there's going to be access control lists built into your network devices and into your operating systems that's only going to allow authorized users to gain access to that information. If the data's not in motion and the data is not at rest, then the data is in use. It's in the memory of a device, and it's being accessed by an application to perform calculations, to look up and gather information, and be able to perform calculations of that information. This is usually stored in the system memory.

It's in CPU registers. It might be in a cache. But it's somewhere inside of this computing device. To be able to use this information it has to be in a decrypted form. You cannot encrypt the data and then have the application be able to perform some type of action to the data if it can't possibly read it. So when you bring this into the system memory and it is data in use, it is almost always in a decrypted form. This means though that the bad guys, if they know where to look, could pull that decrypted data right out of memory and be able to store it, manipulate it, or do anything they'd like with that information.

This is a very attractive option for the bad guys. And if they know that they can't access the data across the network and they can't access the data at rest, then perhaps getting

to the data in use maybe a perfect place to go. In fact, this is exactly what happened in November of 2013 when we discovered the Target corporation's breach where there were well over 100 million credit cards that were then made available to the bad guys. They were not able to see this information go across the network because Target properly was encrypted credit card information as it went over the network in transit.

Target also was storing their information in an encrypted form. So all of the data at rest was also something the bad guys couldn't access. So what the bad guys did was put software on the registers themselves, the point of sale terminals. Those terminals were running a version of Windows. And they simply added some Malware into those terminals that would take the information in memory where the credit card numbers were stored, pull that information off, and store them off into a separate area that then was transferred out of the Target network back to the bad guys.

They knew exactly where to go to take advantage of all of this data in use. And that's why we need security controls not only on our networks, in our databases, in our data centers, and our storage devices, but also on our computing devices itself. So we can protect data in motion, data at rest, and the data in use.

**Tags:** at-rest, certification, comptia, data, in-transit, in-use, security, state

**Category:** CompTIA Security+ SY0-401

### **Permissions and ACLs – CompTIA Security+ SY0-401: 4.4**

A common way to secure files and networks is through the use of access control lists. In this video, you'll learn about ACLs and how they are used for network and file system security

An **access control list** is a set of permissions that are then assigned to an object. You'll hear these referred to as **ACLs or acls**. And they're used on many different kinds of technologies. They're used in firewalls, and switches, and routers, and operating systems. All of these use ACLs to some degree to allow or restrict access to certain parts of the network or to certain parts of an operating system.

An **ACL** is usually referring to a set of permissions and applying that to an object. So things like Bob can read certain files on a file server. Or Fred can access a certain part of the network. They can also be very specific.

For instance, James can access network 192.168.1.0/24 if he is using TCP ports 80, 443, and 8088. You can see that you can build very complex ACLs depending on the type of permissions you need for that particular object.

Many operating systems use ACLs to allow access to files. These are the rights and permissions that you might assign to a user or might assign to a group. So you can apply a set of permissions for the marketing group to be able to access advertising information. But you might restrict that same area of your operating system files to something like the shipping and receiving department.

These ACLs can also be very complex. And you can create very specific controls using these access control lists. Here's an example of an access control list you might see on a network device, like a firewall or router. This is something that shows an access list of access list 1 would deny any traffic that is coming from 172.16.15.2. And there's a mask at the end. And this particular mask means that it's specifying just this IP address.

We're also going to have as part of the same access list a deny statement that denies 172.16.5.3. So if these two IP addresses should never go across the network, these first two statements of this network access control list will deny any access through this device. And you can see the last statement in the access list permits any, which means if you

don't match the 172.16.5.2 or the 172.16.5.3, than anybody else is allowed access through the network.

These are very simple access lists in this particular view. But that should give you an idea of how you can use this top down approach to begin adding different rules that would allow or disallow access to your network or into your operating system.

**Tags:** [acl](#), [certification](#), [comptia](#), [file](#), [network](#), [permissions](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Data Policies – CompTIA Security+ SY0-401: 4.4**

How do you manage your data? In this video, you'll learn the importance of creating security policies around data wiping, disposing of data, and data retention.

One data policy we commonly see is one around data wiping. This is one where we are removing data from a device. This is something the administrator usually initiates by clicking a button or flipping a switch, and removing part or all of the data on a remote device.

Data wiping may also be a policy we apply to hardware that we are retiring. If we have a computer that we are disposing of, or we're transferring ownership of that device, there needs to be a policy in place to find out what you would do with the data on that device. Do you erase everything on the hard drives? Do you remove the storage devices and destroy those storage devices?

In any case we want to be sure that we are not letting any of the data that's currently on that device get into the hands of a third party. These data wiping policies may also be based around your employee onboarding and offboarding process. If someone is leaving the organization, you want to be sure that the data is retained, but you want to be sure that data does not leave with the individual. And if this data is on a mobile device, it may be part of the standard policy that when somebody leaves the organization you're also going to remotely wipe everything on their mobile device.

Other data policies might revolve around the disposing of data. You're gathering all of this information into databases and storing it into your storage area networks, and then how do you ultimately remove that data from the database and dispose of it? There may be legal requirements around what you can keep and what you can destroy, so you'd want to be very clear of what those data policies might be.

Sometimes you're removing data so that you can make room for more data. If you're archiving information on to tapes and you're storing those tapes at a third party or storing them locally, they're going to take up physical room somewhere. And as you archive more data, you're going to be storing a lot more information. So you may want to have a data policy that determines how long you archive that data so that you can remove the old data to make room for some of the new archives that you're creating.

An important part of this data disposal policy should also revolve around Personally Identifiable Information, or PII. People usually don't take well to their private information being stored at all, and certainly not over a long time frame. But if you're in an environment where you need that personal information to be able to perform the function of your company, you have to at least store it on these devices at least temporarily.

And that is the key with PII. That you store it as little as possible. You perform the functions that's needed, and when you no longer need that information, that may be the time very quickly to then dispose of anything that might be personal information. You also might want to destroy information just so it doesn't get into the hands of anyone else.

This is especially important if you're working with very sensitive data in a financial organization, a health care organization. Once that data is no longer needed, you may want to make sure that it's destroyed so that it never moves anywhere outside of your organization.

Just as you have data policies for what you can destroy, you also need data policies for what you plan to keep. There maybe policies that say that you're going to keep versions of programs or data files on a system for a certain amount of time. May want to make sure for instance that all of the accounting department spreadsheets are constantly saved so that you have multiple versions of a spreadsheet available. And if the accounting department needed to go back to a version of last week or last month, you would have a policy in place that allows those particular versions to be automatically saved over that entire time frame.

These retention policies may also be in use so that you can determine how far back you can go with something like backups. If you have a virus outbreak or malware outbreak, you may need to have a number of different backups available going back over as many as 30 days. And it's your data retention policy that's going to determine how much data you're going to store so that you can plan for these types of problems.

You also want to think about the legal requirements around the data that you're keeping. Email for instance is very common to have a legal restriction that maintains that data over years of time. So you have to already have that policy in place, and your processes available to back up your email and have it available for years at a time.

In some industries, certain types of data must be retained over a long time frame. For instance, with financial organizations all of the financial details must be stored over a very long period of time by law. And the data that you're storing usually has different requirements for the storage, for instance, something like tax information or private customer information may require encryption as you're storing this off to these devices. That way if you're storing it on a backup tape, and that backup tape is then lost, you would at least be assured that none of the data on that tape would be accessible to anyone else.

**Tags:** certification, comptia, disposing, retention, security, wiping

**Category:** CompTIA Security+ SY0-401

### **Embedded System Security – CompTIA Security+ SY0-401: 4.5**

We have added technology to almost every aspect of our lives. In this video, you'll learn about the challenges with security embedded systems like **SCADA, HVAC, multifunction printers**, and even our **automobiles**.

On our PCs and our mobile devices we can make changes to the operating systems at any time. We can install new applications. We can modify or update the operating system itself, and really change the entire working environment of that computing device.

But in static computing devices things don't change very much. In fact, there's very little change that would occur on a static device. From a security perspective, this is great. We know exactly what type of protections we need in place. And we know that nothing is going to change outside the scope that we've now built in that static device.

We often see these static environments used when there is an embedded system. These are systems that are created to perform a series of functions. And it's a very specific scope of functions.

If you go into a hospital and there's an intravenous drip meter, that is an embedded computing system inside of that device. Or if you're going to a water treatment plant, and you're looking at the controls that are in use, those are designed, obviously, very specifically for the needs of that water treatment plant.

But just because these embedded systems are there, and they are static types of systems, doesn't mean they will never be updated. There are very often, in fact, firmware upgrades for these that will at least update, or modify, some of the capabilities of that device. But even then, the scope of change is nothing like that you would have for a traditional PC or computer that you would have on your desk. Generally, these are bug fixes or minor changes to the operation of that embedded device.

Two types of very industrial embedded devices are **SCADA** and **H-V-A-C**. **SCADA** is the **Supervisory Control And Data Acquisition System technology**. These are used on very large-scale industrial devices. Another type of industrial device you'd run into is something like an **H-V-A-C** device. Obviously, in the **Heating, Ventilation, and Air Conditioning devices** we also have these embedded systems running to be able to run those particular environmental systems.

Usually there is a PC that manages these devices. And in the case of something like SCADA, you would have a computer that uses the SCADA communication and technology to be able to manage power generators, or refining equipment, or manufacturing devices. These are very large industrial devices. And it is this very specialized SCADA instruction set that is used to be able to manage and maintain these very specialized pieces of equipment.

When we built these devices then they were built with the idea that they'd never be connected to the internet. But obviously that's now a significant concern because you don't want somebody from the outside to gain access to the system that's providing power for an entire city. And that's exactly what is at risk with these SCADA systems.

So obviously, these days, there is an enormous requirement on protecting these SCADA systems. There are laws enacted that will ensure that these SCADA systems are protected. And there are best practices to be able to protect these from the outside.

You generally don't have SCADA systems connected to the internet. You make sure there are firewalls protecting the access, and that the proper access controls are in place so that you can be assured that only the people who need access to these SCADA systems will be the only ones to ever touch it.

We've been printing, and scanning, and faxing for years. And today the technology around this has really improved so that you can combine all of these functions within one single all-in-one device. You may see these also referred to as multifunction devices, or MFDs. Everything that you would need is now in one device. You plug it in and it's able to perform the printing, the faxing, and the scanning functionality all from the single machine.

It's no longer a simple printer, of course. There's some very advanced technology in the hardware. And, of course, some significant software that's running inside of this particular kind of embedded device.

These multifunction devices have a lot of memory inside of them, and software. And it's not uncommon for them to queue up and store print outs, scans, or faxes in the memory of the device. Even after that particular print out has come out, it may still have a copy of that in memory. And someone who knows what they are doing can press a few buttons and have all the contents of memory reprinted. And someone may have access to that data and you had no idea it was being stored on that machine.

There's also, of course, a number of logs stored on this device. So if you are sending a fax, or if information is received from that device, someone may be able to go through that metadata to see exactly who is sending and who's receiving information from that machine.

There's an amazing amount of computing power just within our automobiles. We know that we've had computers in our engines for years and years. And these days, the technology has made itself inside of the car, and we use it for our satellite radio systems, our entertainment units, and our GPS systems themselves.

The technology in the engines themselves has improved through the years. And it's not uncommon to have multiple computers managing different systems inside of your engine. These days you could take your car in and have a firmware upgrade applied to it, to improve the performance or the fuel consumption of your engines.

There's also a side of your engine that's maintaining a bit of telemetry. It's not unusual to have data recorders inside of your systems that are recording the speeds that you travel, the locations you're traveling. And it keeps all of this telemetry constantly stored in your system. And we're just starting to see the legal ramifications of this, since now we're starting to use this data for accident reports, and to show exactly what was happening to a particular device at a particular time of the day.

Maintaining the security of a static or embedded system is a little bit different than maintaining it for our traditional PC environment. But it's still incredibly important to make sure that our information stays private and secure.

**Tags:** automobile, certification, comptia, embedded, fax, hvac, printer, scada, scanner, security, static

**Category:** CompTIA Security+ SY0-401

## **Static OS Environments – CompTIA Security+ SY0-401: 4.5**

We are surrounded by static operating systems, and the number keeps growing. In this video, you'll learn about the static operating systems, iOS, Android, smart television operating systems, and more.

As security professionals, we spend a lot of time securing our desktops. That would be our Windows devices, our **Mac OS X**, or our **Linux devices**. But there are a number of static operating systems that we must consider before truly going to secure the entire enterprise.

Static devices are those where the operating system and the hardware are tightly coupled together. In fact, we look at these devices and often consider them appliances, or standalone units, because it's so difficult to detach the operating system itself from the hardware. In fact, it's very difficult in these environments to install a different operating system because of this tight coupling.

If you need to get updates or upgrades to these devices you always have to go directly to the manufacturer. There's not going to be a third party that makes updates for these devices because they are so specialized and you have that tight coupling between the hardware and the software. This ranges from mobile devices, like the one pictured here, to game consoles, and smart televisions, and many other devices as well.

We sometimes don't even think about these devices as being full blown computers. We just think about the television that we're watching. But behind the scenes, there's an entire operating system with many different functions that are going on.

A popular mobile operating system is one from Apple called **iOS**. **iOS** is an operating system that works across many of the **Apple products**, such as **iPods**,  **iPhones**, and **iPads**. This is definitely a closed operating system. You don't have access to change anything in the OS. You can't make updates to the operating system independently. All the updates and all of the changes to the operating system come directly from Apple.

The iOS operating system was originally derived from Unix. So if you had a way to look under the hood you would see a Unix engine at the heart of iOS.

If you need to get applications to run on this particular platform then you need to get them from Apple's App Store. There's one place to go to get any of the downloads that you would need to run on this device. It is very centralized.

And so developers must submit their application to the App Store so that all of the users can then download that. There's no other mechanism in place to get applications for these iOS devices.

At first glance, it seems that this closed environment would be a hindrance for application developers. But from a security perspective, this closed environment actually helps make things more secure. Since there is a single gateway to the App Store, and everything must be approved before getting on the App Store, it means that the applications tend to be more secure once the users begin downloading them.

Another popular mobile operating system is Android. Android is from the Open Handset Alliance. And Google plays a very large part at leading the direction of where Android goes.

This is certainly a more open model than Apple's iOS. And this is an open-source operating system. It was designed from the very beginning to provide more of an open architecture for these mobile devices.

The application distribution system, then, is not completely centralized. You can certainly go to the Google Play front end to be able to download your applications. But application developers could simply install the software on their web server. And you could download that software directly into your Android device.

Because of this, the Android operating system tends to be more susceptible to malware. Because now the bad guys can fool you into downloading the software directly from the bad guys website.

This doesn't mean that the malware has access to all of the data on your Android device. In fact, the Android operating system was built with Sandboxes in place so that the applications only have limited access to the data. And you must grant them additional access if that's what you'd like to do. This gives the user a lot more control, and limits the scope of what malware could do on this mobile device.

Our new generation of televisions provides us with a number of different capabilities. You may see these smart televisions referred to as connect TVs, or hybrid TVs. And that's because they're more than just a television. They provide us with streaming capabilities for audio and video. We can get video on demand directly from our television, instead of using a separate set-top box.

There's even games and other types of applications that you could run right on your television without needing an antenna or any other input into the TV. You simply plug-in an ethernet connection, or you connect it to your wireless network, and it now has all of these capabilities available to it.

Under the surface these televisions are running a Linux kernel. So as a security professional we have to be concerned about the applications that are running on top of that **Linux kernel**, like **Java**, **JavaScript**, or **HTML5**. We have to keep in mind that the applications themselves are susceptible to tampering and malicious activity. So this becomes another entry point into our environment that we must secure.

This means, if we're using this television in something like a conference room— to do video conferencing— we may want to consider removing or disabling all of the additional smart TV features. That way we can still use the primary display capabilities of the device, but we can avoid having any application perform anything malicious inside of our network.

These obviously are proprietary hardware, with proprietary software running on them, which certainly fits the scope of the static computing environments. We're using these mainframes for very large scale applications. If we need to store large amounts of data, and be able to process that data, mainframes excel at providing us with those CPU cycles necessary to handle these extremely large data sets.

They're extremely reliable. There's redundancy built into the hardware and into the operating system itself. It's not uncommon to have these mainframes continue to run, unaided, for decades at a time.

It's very difficult to find an attack that would have been specifically written for a mainframe. These mainframe operating systems have been around for a very long time, and they tend to be very secure. We also have the luxury that there aren't a lot of mainframes out in the world. So the malware authors aren't going to spend a lot of time developing attack code that's only going to be able to reach a handful of systems.

If there are attacks to a mainframe, these tend to be more on the inside of the network. And they tend to go after very specific types of information. Since so much data is stored on these mainframes, it is a very attractive site for somebody who wants to gather all of that very important data and remove it, or take it somewhere outside of the organization.

One of my favorite static operating environments is that of the game console. This is almost like running a personal computer right next to your television. In fact, for something like Xbox and PlayStation, we really are running versions of Windows and Linux on these devices.

They have storage capabilities. There's advanced graphics display capabilities, a very powerful CPU is inside of these devices. If you weren't running this as a game device, it would make a very good file server.

Many people, in fact, will root or jailbreak these devices so that they can use them to do other things as well. These devices were not designed to work this way, and the manufacturers certainly don't recommend you do this. But if you know what you're doing you can build a system that provides you with some additional capabilities using this very powerful hardware.

These devices are very network oriented. They use the network to be able to connect to other gaming systems, and to be able to download games and updates to the operating system of the game console. Because of this, this might not be the best use on a corporate network because you don't have control over patching, and being able to secure the operating system itself on these game devices.

**Tags:** [android](#), [certification](#), [compTIA](#), [embedded](#), [game console](#), [ios](#), [mainframe](#), [security](#), [static](#), [television](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Mitigating Risk in Static Environments – CompTIA Security+ SY0-401: 4.5**

Even static environments require security. In this video, you'll learn about layered security, network segmentation, tcp wrappers, application firewalls, and firmware version control.

Static operating environments create a few additional challenges for security professionals to be able to mitigate risk. In this video, we'll talk about some of the techniques we use, not only to mitigate risk on our traditional operating systems, but we'll talk about how we can change these just a bit to take into account the static operating environments.

One thing that doesn't change when you're working in a static operating environment is the need for layered security. We want to have what we've always called defense-in-depth. There's never just one single security device in place. We want to have different strategies and different processes in place so that we can every step along the way prevent that particular security risk from getting through our system.

The security controls themselves should also be very diverse. In fact, it might be a little different than the picture that shows exactly the same hurdle, one after the other. Instead, we might have hurdles that were different sizes and different widths to prevent the same type of attack from getting through our system. We would want to have a different firewalls and different IPSes.

And we might want to have different processes, all working together, to have this defense-in-depth. We want to also avoid any single points of failure. If the bad guy can bring down your firewall, there's potential then for them to go right through all of your other security systems. So you might want to consider multiple firewalls, redundant intrusion prevention systems, and even multiple management systems so that if one management system goes down, you can still be alerted and get messages should any security concerns arise.

One very common defense against the bad guys would be to segment the network into different pieces. We might have one logical section of the network for internet traffic, another one for our DMZ. There may be a storage network. We might also have a separate network just for the management of the network. And our corporate environment may have another section all unto itself.

The idea here is to limit the impact of a breach in the network. If someone does make their way to the DMZ, we've already segmented out the rest of the network. And at least that particular breach would not be able to impact other parts of the organization. The segmentation may be physical. We might have separate switches, and separate routers, and separate security infrastructure for each individual section of the network.

Or we might set this up to be more logical. We might have separate zones and a firewall. We might separate out rules based on IP address, or destinations, or sources. By doing this, we can create a logical separation without needing additional hardware or infrastructure to be able to protect these parts of the network.

We want to think about how we want to separate out the network, and then create separate security policies based on these separate zones. So, for instance, you might require that nobody has any personally identifiable information in the DMZ. And you want to be sure that if any credit card information is transmitted that it never goes across the network, and it's always encrypted when it does travel anywhere else in the network.

We created TCP wrappers as a very early form of application control. This allowed us to put a wrapper between the network and the service that was running over these network packets, to give us a little more visibility into the application that was going across our networks. And we used access control lists then to be able to manage whether certain types of application traffic could go over our network or not.

Today's application firewalls take a complete holistic view of the traffic patterns that are going across the network and are performing deep packet inspection to be able to truly recognize all of the different applications that are going over the network. We can even get some very specific types of application definitions.

For instance, we can recognize general Facebook traffic, which is different than somebody trying to post to Facebook, which is also very different than somebody trying to chat on Facebook. Each one of these is seen as a separate application. And we can set controls in our application firewall to allow or disallow all of these very specific application types.

These application firewalls can also find very specialized applications like SCADA, where you really need to protect some of this very large industrial equipment from being accessed from anywhere on the outside. These embedded operating environments were built to perform a particular task. There's very specialized software that works with very specialized hardware. And they're always going to perform that task and nothing else.

Because of this, you don't tend to have a lot of updates to the operating system or any other part of this embedded working environment. Some of these embedded systems were never designed to be updated. There's no media that you can plug into the device. And if you do need to, for some reason, perform a firmware upgrade, you would have to completely remove the hardware and bring in a replacement unit that already has the new firmware on it.

Even these systems that do allow for updates aren't necessarily designed to do things automatically. It's usually a manual process to perform the upgrade to these devices. They don't become part of your windows domain.

There's no overall management software that pushes out updates automatically. It's more of a time consuming process. And because of that it may be less of a priority, which is

certainly a concern when it comes to dealing with security in these embedded operating environments.

**Tags:** certification, comptia, embedded, firmware, risk, security, segmentation, static, tcp wrappers

**Category:** CompTIA Security+ SY0-401

### **RADIUS and TACACS – CompTIA Security+ SY0-401: 5.1**

A well-designed network will use a single authentication method for all services. In this video, you'll learn how RADIUS and TACACS can be used to centralize the authentication process.

**Remote access administration** is a key component of today's enterprise networks, and it's something we even take advantage of when we're using these resources across the internet. If we're logging into Facebook, if we're logging into Google, if we're logging into Yahoo, we could be connecting to one of many different servers that those organizations might have anywhere in the world. And somehow we're able to put in our user name and password and magically we gain access to those resources.

That's not because Google and Yahoo and Facebook have copied their entire user database to every single server they happen to have. What they have is a method that goes back to a single authentication server, and that authentication server is able to give you access and rights and permissions to those particular resources.

You might log in in different locations. You might be logging in from your desktop. You might be logging in on a **VPN tunnel** from somewhere outside of your network. You might be logging into a router to provide administration, and you simply use the same user name and password that you would use for every single one of these. It doesn't matter. That's one of the nice things about this centralized management is you don't have to remember a lot of user names and passwords. You use the same authentication credentials whatever you might be doing.

This is an important security concept of **AAA**. It's **authentication**, **authorization**, and **accounting**. This AAA concept is one where it's able to check the credentials that you're using with your user name and password. It's able to provide the proper access to the network based on who you might be. And it's also able to track when you logged in and when you logged off, and perhaps other things in between. Those concepts become extremely important when dealing with security. And, of course, they're going to be part of this remote access authentication.

One very common way to gain access to a network and get authenticated is through something called **RADIUS**. **RADIUS** stands for **remote authentication dial-in user service**. The first **RADIUS RFC** was our **RFC 2058**. The most current version is RFC 2865. You have people in your environment that are logging in remotely from over the internet. They may be people on wireless client devices or they may be your users inside of your network that simply need to authenticate in their normal way.

In each one of these devices, your remote access server, your wireless access point on your intranet based devices, all of these have RADIUS clients on them already. And when you log in— let's say these people out on remote access are logging into this remote access server— you've previously configured this remote access server to say, if anybody ever needs to authenticate, let's use the RADIUS protocol and let's communicate back to this centralized AAA server to be able to authenticate those people.

So when I try to log in remotely, the first thing I'm prompted for is a user name and password. I might also be prompted for another piece of information for additional two-factor authentication. A random number or some other type of information. I'll provide all of that to that client that pops up asking for those credentials. That is sent to this AAA server. And RADIUS usually uses UDP over port 1812 by default to provide that access.

The AAA server checks my user name, checks my password, maybe checks that two-factor authentication information. And if everything is legitimate, it logs me in and makes a note of when I entered the network. And then when I log off, it's also going to make a note of when I logged off. It's that process that allows me to centralize this. Doesn't matter if I'm coming in remotely. Doesn't matter if I'm a wireless client. Doesn't matter if I'm on my local intranet. Everybody's able to get the same type of authentication using the same user name and the same password that they always use.

One option to **RADIUS** is something called **TACACS**. **TACACS** stands for **terminal access controller access-control system**. And it has been around for a long time. The original TACACS standard is created in **RFC 1492**. It was written up. And this was originally created to control access to the dial-up lines to **ARPANET**. So this is before the internet really ever became the internet.

This is one where you wanted to restrict who had access to these dial-up lines, so these guys got together and created a remote authentication protocol that would do that. Well, later on, there was another type of **TACACS** called **extended TACACS or XTACACS**. This is something Cisco created that extended the capabilities of TACACS. It's proprietary to **Cisco**, but it's one that allowed Cisco to add additional support for accounting and auditing.

These days, you don't tend to see TACACS or XTACACS. Usually you see TACACS+. It is the most modern version of this. It's also Cisco proprietary, but it's one that adds additional authentication requests and response codes. You just have to remember that it's not backwards compatible with these other TACACS formats.

One of the things that a lot of administrators like about TACACS+ is that TACACS+ uses TCP over port 49 to communicate, and that's a little bit different than RADIUS that uses UDP. And many administrators feel that that TCP connection oriented and reliable protocols is one that has a little bit more advantages over RADIUS.

But in the big picture, both RADIUS and TACACS+ are performing similar functions. Usually it depends on the type of network or the type of devices you have on your network, and what they expect to use to be able to perform the centralized authentication.

**Tags:** [authentication](#), [certification](#), [comptia](#), [radius](#), [security](#), [tacacs](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Kerberos – CompTIA Security+ SY0-401: 5.1**

Many organizations prefer using a centralized and secure authentication protocol. In this video, you'll learn how Kerberos uses many different encryption points to maintain the security of the authentication process.

If you've worked in a number of enterprise environments, then you're probably familiar with the term Kerberos. Kerberos is a network authentication protocol. It's one where you can authenticate one with the system, and then you're trusted by everything that's in that entire infrastructure.

The idea is that you're being able to authenticate with a central server, and then also authenticate with everything else. But not only are you authenticating yourself to the server, you can also be assured that you are talking directly to the server. So there's mutual authentication here. And what that does is help anybody who might be in the

middle of this conversation who wants to inject particular pieces of information. Or maybe they'd like to replay this information later on; they would not be able to do that.

This is a standard that was created back in the 1980s, so it's been around for quite some time. It was created at MIT. And there's an RFC for this. RFC 4120 gives a lot of details about the Kerberos standard.

Microsoft started using Kerberos with Windows 2000, and they use Kerberos 5.0 which is an open standard. This makes it very easy, also, for other operating systems to be able to authenticate into the same Kerberos-based environments. So even if you have a Linux or you have Mac OSX, it doesn't matter. Everybody ideally can talk back to the central Kerberos system and be able to authenticate onto the network.

**Kerberos or Cerberus** is a mythological creature. This is the three-headed dog of the underworld. Its job was to keep people from escaping across the **River Styx**. And a three-headed dog would certainly do that for me. But it had three heads for a reason. There's a reason we call this **Kerberos**. That's because there are three components to this.

One, is that you have a key distribution center. You'll often see this referred to as a **KDC**. This is something that is vouching for the user's identity. We're providing user name, password, and other authentication information, and we're getting tickets that we can use later on. We're going to go through that process. This runs either on tcp/port88 most commonly. You could see it on udp/port88, as well.

The second head of the three-headed dog is the **authentication service**, which does exactly that. It authenticates us and provides us access to the network. And the last head is the ticket-granting service that provides us with tickets. In Kerberos, tickets is a pretty important thing. You're given tickets that will gain you access to resources on the network.

Before you can gain access to resources, you first have to authenticate yourself with the key distribution center. This is the authentication service that's going to provide us with everything we need to gain access to other resources. And this is a 2-step process.

We'll start down here with our device, our client, our principle we call it. And it's going to talk directly to the key distribution center up here. There's an application server down here that we would like to access, but we're not able to do that yet until we complete this authentication process. So we're going to send a login request from our device to the key distribution center. And we're going to send this encrypted with the date and time on the local computer. And we're going to use our password hash as the key.

Now we don't send the hash to the key distribution center. This authentication service that's up here already has our password. So it knows what it should be expecting to use as that key, and the next process is going to take advantage of that. But it's important to know that this entire process is encrypted and it's very, very secure.

When the key distribution center receives this encrypted package that's the authentication request, it decrypts it with what it knows to be the client's password hash, has a look at it, and it makes sure that the time frame that was encrypted in there is somewhere within a five-minute period. So this is very time sensitive. Inside of that, it checks and makes sure. And if that's all legitimate, it sends back what is called a **Ticket Granting Ticket**— sort of an odd name.

It's a ticket that's going to allow you to get other tickets. It's a pretty important ticket. In that ticket, it's going to have a client name, an IP address, some timestamp information, and a validity period. So that this is only going to be good for a certain amount of time. After that, you'll have to re-authenticate to the network. This is also encrypted. You can see there's a key associated with this. It is encrypted with a secret key for the key distribution center, which means the client here is not going to be able to decrypt this.

This is pretty important because we want to be sure, if we're going to authenticate to a third party, that our authentication is going to be trusted.

And if we know the key distribution center is the thing that has the private key, and that's the only thing that has that private key, we can be assured that that particular piece of information is going to be well-protected when we present it to our application server.

Another piece of information that we get when we authenticate to the key distribution center, is a Ticket Granting service session key. And this is used to encrypt the communication between our ticket granting service that's up here, and our client. And we're going to use that session service key again. That will be very useful. Again, every part of this process is encrypted, and in this way, we can be assured that nobody is able to look into what's going on. Nobody's able to break open some packets and see information that we may be authenticating— user names, password, or anything else that might be in here.

It's interesting to note that that particular session key is encrypted with the user's password hash, so we will be able to decrypt that, and look inside of it and see all of the details inside of that piece. So notice that we're getting private keys from one place, private keys from another place. There's a lot of distribution here. It's a very complex process to make this happen. Fortunately when you're authenticating on to the network, you just put in a user name and password. You've no idea this complex encryption process is going on behind the scenes.

Now that I have been authenticated, I would like to be able to communicate and use some resources on the network. Specifically, it would be great if I could use this application server down here. But I've never spoken to the application server. The application server itself has not communicated to the Ticket Granting Service. It has no idea that I'm on the network. What I have to do is get a ticket from the Ticket Granting Service that's going to provide me with access to that application server.

The first step we'll do is from our client workstation. I've got that encrypted Ticket Granting ticket, and I'm going to attach to that the name of the service that I'd like. And I'm going to send that to the **Ticket Granting Service**. And I'm going to also send a time stamp client ID that I've encrypted with my session key. Again, we've got all this encryption that's taking place, and the Ticket Granting Service can certainly decrypt its own ticket. And since it knows the hash that I'm using as my password, it can also decrypt the **TGS** session key, as well.

If this set of information that we provided to our Ticket Granting Service looks good, then we're going to get a couple of things back from the Ticket Granting Service. The first one is a service session key that we'll be able to use with the application server. This is a key and information that is encrypted with the session key that we know about. So we'll be able to look at that information once we receive it off of the network.

We're also going to get a ticket. And this particular ticket is one that is going to have user information and session information key, and it's going to be encrypted with a private key from the application server's secret key. So this is a secret key that only the key distribution server and the application server know about. I don't have any access to that private key, so therefore, I'm not going to be able to decrypt that information and see what's inside of it. It's going to be protected. And in our next step, we're going to provide that to our application server. That's why it's so important that that information stay absolutely private.

And that's exactly what we're going to do in our next step. We're finally going to talk to our application server, and we're going to send along that private package that we received from our key granting service. This is our encrypted service ticket, and it's going

to be encrypted with that private key that we know nothing about. So we want to be sure we're providing exactly the same key to the application server.

We're also going to provide an authenticator. It's going to have a timestamp in it, and it's going to be encrypted with our service session key that we got. That way we're able to send all of this encrypted information, and on the other side, our application server should be able to see all of this. Our application server is going to look at this service ticket that was provided to us that was encrypted and we couldn't look inside of it. So it's now finally going to decrypt it with this private key, and make sure the information in there looks OK.

It's also going to look at the authenticator we sent it and make sure that the session key that we have with our passwords on it matches what the application server would be expecting with that. There's also an optional process— the application server may send the timestamp back to the client encrypted with that service session key. And because we're sending that information back now, we are really checking to see if there's a man in the middle. We want to prevent anybody from sitting in the middle and looking at this information. We also want to be sure that nobody can replay this information later, to try to gain access to these resources.

After exchanging all this encrypted information with our key granting service, we are now talking directly to the application server. The application server says that information checks out, then you now have access to those resources. This Kerberos process takes place any time we need to gain access to the network and authenticate for the first time, and every time that we need access to yet another resource.

So if you go into, for instance your Microsoft Windows Active Directory infrastructure, you look in the security logs, you'll see a lot of session information where people are authenticating into the Kerberos system. And you'll see every time someone gains access to any of those resources. This Kerberos technology allows us to be able to authenticate and provide access, regardless of where you might be on the network, or really even any type of operating system that you might be using.

**Tags:** [authentication](#), [certification](#), [compTIA](#), [Kerberos](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **LDAP and Secure LDAP – CompTIA Security+ SY0-401: 5.1**

Most organizations maintain a centralized database that's used for authentication, user identification, and many other purposes. In this video, you'll learn how LDAP and Secure LDAP can be used to efficiently manage these large user databases.

In our computing environments, we have large directories of information. We probably have a list of everybody in our organization. We have their location, their telephone numbers, their email addresses. And this very large index is called a **directory**. There are protocols that then allow us to work with these large directories, and one of those is called **LDAP**. And this stands for **Lightweight Directory Access Protocol**.

This **LDAP protocol** is very standardized and it uses a standard called X.500. This is a specification that was written by the International Telecommunications Union, who obviously have a very big need to be able to interact with different telephone directories. The original **LDAP** was simply called **DAP**, the **Directory Access Protocol**. And it ran using the OSI protocol stack, a protocol stack we don't often see running any longer.

These days we use a lightweight version of **DAP** called **LDAP**, and it uses **TCP/IP** to communicate over **TCP port 389** and **UDP port 389**. If you need to update or modify things in a directory that is X.500 compliant, then you're going to use the LDAP protocol. And it's very common to see this on all of our major operating systems. If you're running Active Directory with Microsoft Windows, you're running Apple's Open Directory, or you're

running Novell eDirectory, then you're using LDAP for each one of those directory services.

X.500 has a very particular way of defining what's in the directory. It does this using distinguished names. It puts information in pairs where you have an attribute, an equal sign, and then what the attribute value happens to be. The most specific attributes tend to be specified first in the big list of these distinguished names.

For instance, this is the distinguished name for a web server called **WIDGETWEB**. In fact, it has an attribute at the beginning of **CN**, and I have a number of the popular attributes here. **CN** stands for the ***common name of that device***. So the common name is **WIDGETWEB**, which is the web server that we're specifying here. It is in an organizational unit of Marketing.

It is in an organization called Widget located in a locality of London. It is in a state of London. The country is Great Britain. The domain component— this is used a lot when you're trying to describe the components within this— are widget and a domain component of com. This describes this web server as being the one used on Widget.com.

So you see all of these working together allow us to provide a very specific representation and directory of this particular web server. This modular nature of the distinguished names in LDAP allows us to build a tree of different devices. And a good example is the picture that we have here.

We've broken the tree out into two pieces. We have container objects that store other objects within it. Objects such as the country, the organization, and the organizational units. Within these containers, we have leaf objects which are the individual devices, such as users, computers, printers, and the files. And you can see how you might build a tree based on this.

For instance, we have the country of Great Britain. Inside of Great Britain is a Widget object. You also have a container object within the Widget object of Marketing, Accounts, and Mis. And within Marketing, you then have separate devices such as the Webmaster and such as the WIDGETWEB. All of these can now be easily defined and you can put the objects into the proper object container, or you can define exactly what container they belong to.

There are number of additional security layers that you can have within **LDAP**. One is called the ***simple authentication and security layer, or SASL***. And this is in **LDAP** version three. There's different types of authentication that this would allow us to have to these directories. The first would be no authentication, which means that anybody gets access to the data.

You can have anonymous users that would gain whatever type of access they would need to have in that particular data store. There might be simple authentication, where the client is providing a distinguished name and a password and that then provides the access to the directory. Or you could have a simple authentication and security layer where you're integrating with something like Kerberos or TLS to provide an additional layer of security on top of SASL.

Once someone is authenticated into the directory, they generally would have one of two levels of access. You would have a read-only access, where you could simply query the information. And then you might have a read-write access, where you might update the information. It wouldn't be uncommon to allow anonymous users to have read-only access, but have your normal users and administrators have read-write access to the directory.

One important consideration regardless of the authentication type is that you are limiting the access to this information. So if this is an **LDAP database** that is accessible from

outside of your network, you may want to consider putting a firewall in place to prevent unauthorized users from trying to authenticate to your LDAP information.

Another security layer that can be added to **LDAP is LDAPS**. This stands for **LDAP over SSL**. The **LDAP** protocol itself sends all of this information over the network in clear text. And obviously, it's very easy to be able to retrieve these packets off of the network and view that plain text information. We want to therefore be able to encrypt this information, and the common way to do this is using **LDAP over SSL**.

This is using **SSL**— or what is now called **TLS**— to encrypt the information as it goes over the network. It's very common to see LDAPS being used in Microsoft environments. The Active Directory database can be accessed via these LDAP protocols, and instead of using TCP port 389 and using LDAP in the clear, it's very common to use TCP port 636 that's connecting using LDAPS.

If you do have a policy in place that restricts any type of in the clear LDAP communication, then you may want to firewall LDAP ports 389 just to make sure that the only protocol that's going to be using this LDAP directory is LDAPS over TCP port 636.

**Tags:** authentication, certification, comptia, ldap, ldaps, secure ldap, security, x.500

**Category:** CompTIA Security+ SY0-401

## **SAML – CompTIA Security+ SY0-401: 5.1**

You've probably visited a web site that provided authentication using a 3rd-party service, and it may have been using the **Security Association Markup Language (SAML)** to accomplish this. In this video, you'll learn how SAML can be used to authenticate to one set of resources by using a completely different authentication provider.

Now that we use all of these different services in the cloud, it becomes a little bit cumbersome to create new security authentication every time we need access to a particular website. And you may have seen on a website that instead of creating a brand new account, you can authenticate to this website, if you have credentials on a third party website.

It's very common to use a method like **SAML**, which stands for **Security Association Markup Language**, to be able to provide this authentication mechanism to one site without that site having any of your private authentication information. There's obviously fills an important need. Authentication is such an important consideration for these services in the cloud, but of course, creating a separate authentication mechanism for every single website can be tedious for the end user.

And for the service provider there's a number of security concerns for having all of this authentication information. Instead, using SAML allows the service provider to provide the services and an authentication provider to handle all of the authentication parts.

There are generally three pieces that are needed to create this association for authentication. One is the service provider. This is the person that's providing that capability that we need access to. The other is, of course, you the client. You need to gain access to this service, and you're usually doing this in a web browser.

And lastly, we have somebody that has all of that authentication information. In SAML we refer to them as the identity provider because they are containing all the identities and all of those login credentials.

Here's how the flow of this authentication works. We are the client, and we need access to a resource server. So we're going to access the application, and the application will then need us to authenticate. So it sends back a signed and encrypted SAML request, and we're going to take that request and send it on to the authorization server.

At that point we're going to provide our normal credentials to this third party authentication server, and that authorization server is then going to decide whether that authentication is correct. If the authentication is correct, it sends you back a token that has been digitally signed by the authorization server.

You then present that SAML signed token to the resource server. The resource server, of course, trusts anything that is digitally signed by the authorization server, and therefore, you then gain access to whatever resources you need on the resource server. This practice allows you to have a single login on an authorization server, but be able to access a number of different resources, all from different third parties.

**Tags:** [authentication](#), [certification](#), [comptia](#), [saml](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Identification, Authentication, and Authorization – CompTIA Security+ SY0-401: 5.2**

The foundation of access control is based on the three major tenants of **identification**, **authentication**, and **authorization**. In this video, you'll learn how each of these is used to maintain the security of our networks.

Identification is the process of associating a user with something that has occurred on a server, on a network, or with some other resource. You need to know who accessed a particular file. You need to understand who just logged into a server. That is the process we call **identification**.

This information is almost always logged. And there's usually a username or some type of very unique identifier assigned to that particular function. This identifier could be something like a name of a person— maybe their first name or their last name.

In Windows, there's something called a **Security Identifier**, or a **SID**. And that is something that is assigned to every user on a device. And it's something unique for every user on that device.

We could use something like a smart card, or a certificate that we might carry with us, and that would certainly identify us uniquely. There could be also biometrics in use. Maybe we use something like a fingerprint, or a retina scan, to be able to identify us uniquely. Or certainly something like a verification card that has our picture and our personal information on it. That may be something that we provide to an end user, or to a third party, that does indeed say that we are identified as this particular person.

It's not enough to say that you're a particular user or have a label associated with yourself. You also have to prove that you are that person. And that is the process of authentication. The authentication process means that you're going through some extra steps to prove you are who you say you are. We can't just take at face value that you happen to be that user. We need some other type of proof to be made available.

This proof would commonly be something like the combination of a username and a secret password or passphrase. That combination of things together would help prove that you are who you say you are because no one else has that combination of information. But you might also want to add additional authentication types to that. Maybe we do provide biometric information, or provide a pseudo random key generation that is something that you have to have with you when you're authenticating to these resources.

Once you have authenticated to these resources, and we believe we have identified you successfully, now we have to provide you with authorization. This is the step that defines what rights and permissions you have to these particular resources. We need to define this as, perhaps, your name of the user. Perhaps you belong to a certain group of users and therefore you have certain rights and permissions available.

Ultimately, it's access to the resources that is the important part. And these resources may be files, or directories on a file server, or it may be what you can access on the intranet of your network. This is all defined by these rights and permissions that we're assigning based on your authorization to the network.

There also needs to be a way to ensure that these policies are enforced. Now that we know who you are, and you've authenticated to the network, we need to make sure the authorization will provide the limits and the access that you need, based on who you are. This is usually something that might be defined in policies in a firewall, or an access control list on a file server.

## **Authorization and Access Control – CompTIA Security+ SY0-401: 5.2**

We use many different kinds of access control to secure our networks. In this video, you'll learn about discretionary, role-based, and mandatory access control models.

The idea of access control revolves a lot around authorization. And with authorization there is policy enforcement, which means that if a user is trying to access a resource that we've given them exactly the right permissions they should have to access that resource. There's also policy definition, which means we have to define what rights people get to have access to those resources. And there's a lot of different ways to do this.

We can have these access control lists. And we can configure these to be a discretionary access control, a role based access control, or a mandatory access control.

With discretionary access control, the person who owns the resources in complete control of who might have access to that resource. It is very, very flexible. But from a security perspective it's very weak. You have the person who owns that information now making security decisions on who gets to access it. And one small slip from the owner could give full control to everybody, and not really mean to do that.

A step up from that is role-based access control. This is where we are defining what access you may have to a resource based on the role of the user. So you may be a group of people, you may be in a department, you may be part of a team of people, and we'll say that if you're on this team you have the ability to read these particular files.

This role-based access control also gives you a little more control over what's going on. It scales a little bit well. In Windows we use groups to be able to accomplish this. So you would add all of these different users to a group and you would simply assign rights to the entire group at one time.

We often see mandatory access control used in government type environments, where there are security clearance levels. And everything that you would need to access would have a label associated with it. Every document, every printer, everything would be labeled with secret, or top secret, or code blue. And your account would be given a certain level of access.

Your access may be able to access top secret, which means you can access anything labeled with top secret or secret, because that's one level underneath. But you would not be able to access anything that was code blue. It's an interesting approach to being able to control what's out there, and make sure that only the certain people associated with that level of clearance would have access to those resources.

You may hear access control referred to as rule-based access control. And this is more of a high level way to describe the way that rights and permissions are given out. One type, or two types, of rule-based access control is role-based and mandatory access control, because those access control methods are determined by the system. They're not determined by individual users setting this up.

This pre-defined rules and pre-defined processes you have in place have given access to certain people based on a group they might belong to, or based on a particular security level. And it doesn't matter what a user wants, or what somebody else would want to reconfigure, your systems are going to make sure that everything remains as secure as possible.

Another access security concept, extremely important, is one called an implicit deny. We use this a lot there firewall module, and it makes perfect sense. That's one where nobody

has access unless you explicitly give them access. If you don't give them certain access, by default, they are implicitly denied from accessing anything. It's a very important part of what we do, not just firewalls, but with all of these other access methods that we have.

Since many of our organizations are also 24 hour shops, there may be a need to set rights and permissions based on the time of day. So our firewalls, and our operating systems, and some of the other devices that we use allow us to set different types of access control depending on what time of day it might be. And this is important if it's an organization at night that needs to run a lot of backups, or you want to turn off certain access during the day, you can implement that into the operating systems you're using or into the firewalls that you might have.

**Tags:** certification, comptia, discretionary, implicit, deny, mandatory, role-based, rule-based, security, time of day

**Category:** CompTIA Security+ SY0-401

### **Single-factor Authentication – CompTIA Security+ SY0-401: 5.2**

Most of our day-to-day authentication uses a single-factor to provide access. In this video, you'll learn about the different factors of authentication and some of the challenges around using single-factor authentication.

When we talk about authentication factors, we're talking about one of these things. It may be something you know. It may be a password, it may be a personal identification number. It may be something we are typing in or something we know about that we can tell this computer system about that would help prove that we are who we say we are. It might also be something you have. You might have a smart card with you that you physically put into a computer. You might have a token the generates pseudo random numbers, and you're prompted to put in what the latest number is on your token.

And that would ensure that you have something physical with you. So that must be you because you have this physical device that nobody else could possibly have. The other authentication factor we often see is something that we are. It's a biometric. It's a fingerprint, it is a handprint. It's an outline of your hand or your fingers that you would use to put onto a system, and maybe then also put in a personal identification number. You get to choose one of these factors in single factor authentication to provide people with access to the network or access to resources.

Most often, the type of factor we're using for single factor authentication is your password where you have a secret word that only we know about. We'll put in our username and then add that secret password or passphrase into the computer. Now, what's interesting about this is that our username generally isn't something that's private. But it's not something you would want to share with people either. If they have half of the equation they might be able to guess your password.

That's why in a number of organizations you don't get your first initial and last name as your username. You get a bunch of numbers all put together. And if you were to look at that number you'd have no idea who that person was. And that's just another layer of security put on top of everything else to try to prevent somebody from guessing your username and password combination. This password or passphrase is usually a set of numbers.

It's some special characters. It's probably a combination of all these things, some uppercase and lowercase. You want to be sure your password is as strong as possible so people don't guess it. And you might also be required to type in something like a personal identification number. That's another type, another factor of authentication. And this might also have some personally identifiable information, or PII associated with it.

Especially if you need to reset your password. It may ask for what you were high school was, what is your full name, what is your address, what is your social security number, that may be a bit of information that's very personal to you that you'll be able to share with the computer system, or share with the management of the security in your environment to let them know that this is really you're trying to log in or trying to reset your password. Single factor authentication is very easy to implement.

But there are a number of challenges associated with it. And you can imagine if the only thing between you and resource is a password, becomes now very easy for the bad guys to gain access to what's on the network. They'll look over your shoulder as you're typing a password in, or they'll guess your particular password and gain access to resources. And between them and everything is just that password. Nothing else is stopping them from getting into the network.

And the bad guys are getting very good at getting their hands on these passwords. They'll send you an email. The email says your Google account has been compromised, you'll need to log in to verify and reenable your account. And you click a link, and it pops up to something— looks just like Google's login page. But it isn't, it's the bad guy's login page. And you type in your username and password. And now, they have your credentials to get onto that system.

Again, a very big limitation of a single factor environment. Many passwords, of course, very, very easily guessed. We've talked about this before in previous videos that too many people are using the password of password, or 123456, or cookie. It's very, very simple. You can go through a top 50 of the well known passwords, and you'll probably get a pretty good hit on what somebody might be using as their password. And lastly, you want to be sure you don't reuse these passwords.

If your password on Google Mail is one thing, you we use a different password for your domain login, and a different password to log into your banking account, for instance. In 2011, there were some breaches in Sony and Gawker that provided a list of usernames and passwords out to the world. And we found that in 88 emails that were the same between them, 92% of them had the exact same password. We need to get out of this habit of using the same password over, and over, and over again.

Because if the bad guys get a hold of just one of your passwords, they'll now be able to access any of those accounts wherever they may be.

**Tags:** authentication, certification, comptia, security, single-factor

**Category:** CompTIA Security+ SY0-401

### **Multi-factor Authentication – CompTIA Security+ SY0-401: 5.2**

If you want to secure your authentication process, then you'll probably implement some form of multi-factor authentication. In this video, you'll learn how to secure your authentication by using something you are, something you have, something you know, something you are, and something you do.

When we're authenticating to a resource we may be using multiple factors of authentication. We categorize these factors as something you are, something you have, something you know, something you are, and something you do. These might be very expensive methods of authenticating, perhaps using separate hardware tokens, or it's something that might be less expensive, like an application that would run on a smartphone.

Something you are generally refers to something like biometric authentication, so we would be using a fingerprint or a voiceprint or an iris to be able to really identify that you are the person who you say you are. The process of capturing these biometrics from the

beginning is not taking an actual picture of your fingerprint, it's making a mathematical model of your fingerprint or your voice print or whatever you're using, so that later on it can compare your fingerprint using this exact same mathematical model to see if the two things match.

These types of things are difficult to change. It's not often that we would change out a fingerprint, and our voice tends to be exactly the same day after day, week after week. But these processes are still not foolproof. We want to be able to consider using these in very specific instances and perhaps combining them with other factors of authentication as well.

An authentication type I use often is something I have. This means that you have something with you that will help identify you as an individual. This may be something like a smart card. This is something you might slide into or get close to a particular resource reader, and it may then also require you to input a PIN, so that somebody couldn't simply steal your card and gain the access. We're going to add additional layer of security to your smart card.

Another piece is a USB token. There is a certificate on the USB drive that you must insert, and that certificate would then also require something like a PIN to be able to gain that access. Or there might be something like software tokens or hardware tokens, where you are presented with a pseudo random number, so you not only have to provide your username and your password, but you also have to put in whatever number happens to be listed on this particular software token.

Another way to do this is with our telephones. Once we put in our username and password, the system may send us a text message, and we then have to repeat back into the system what was listed in that text message. This is just another way to help prove who we are based on something we might have with us.

A very common and very inexpensive factor of authentication is something you know. This would be something that you've got in your brain. It's in your head. A password is a very good example of something you know. It's a secret phrase or secret word or string of characters put together.

Another example of something you know is a **PIN**. That stands for **personal identification number**, and it's usually associated with an **ATM card** or a **smart card** or some other device, so that you're combining both that device and the special personal identification number to help link those things together.

A third piece of something we might know may be a pattern. This is something you see on Android devices, for instance, where you can lock the screen and then unlock it, if you happen to know the right pattern to move your finger around on the screen. This is a little bit different than knowing a word or a pass phrase or a set of numbers. You have to now remember what that particular pattern was, and then repeat that every time you want to authenticate.

A relatively new method of authentication is something like somewhere you happen to be. This is all based on your location whenever you're trying to authenticate. This would check to see what geography you happen to be in and then allow or disallow access based on that information. One way to do this might be with your IP address. In many cases, at least with IPv4, we can identify what country an IP address was originally assigned to.

So that if we know that you're logging in from an IP address that's located in the United States, we might allow you to continue the authentication process. But if that IP address is trying to authenticate from an IP address that was registered to China, we might automatically restrict any logins from those IP addresses.

You can these days combine this with even additional services, like location services, on our mobile devices. This would really give you some very specific geolocation information, and then you can provide that information then up to the authentication mechanism, and they'll know that you're standing in the front door of where you should be when you're trying to authenticate into the building.

The last factor of authentications that we'll look at is something you do. This is your own personal way of doing things. Everybody has a certain way of signing their name. That is certainly something that we do that's very unique to us. Handwriting analysis is a very common way to do this since everybody has a different style or technique of signing their name.

This might also be something like a pattern of typing. Whenever you type in your password, there might be always a very similar structure you have to that typing, and that might identify you as somebody very unique. This is very similar to biometrics in a way which is something you are. In this particular case we're taking it a little bit further into more of an artistic level and defining it as something you do.

**Tags:** authentication, certification, comptia, multi-factor, security, something \_\_\_\_\_ you are, something you do, something you have, something you know

**Category:** CompTIA Security+ SY0-401

## **One-time Password Algorithms – CompTIA Security+ SY0-401: 5.2**

A useful security authentication technique is the use of one-time passwords. In this video, you'll learn how one-time passwords are implemented and the differences between the **HOTP** and **TOTP** algorithms.

If you've ever authenticated to a resource using multiple forms or factors of authentication then you've probably used a username, a password, and probably some type of one-time password. In this video we'll look at a couple of different ways to provide this one-time password functionality.

One-time passwords are passwords that we use a single time, and we never use them ever again. These passwords may be useful for a single session, or they may really be used every time we want to authenticate. And then, whether we get the authentication correct or incorrect, we're never going to use that one-time password again.

One common way of providing this one-time password is through something called **HOTP**. The stands for **HMAC-based One-Time Password algorithm**. And it uses a keyed-hash message authentication code, or an **HMAC**. This message authentication code is something that's going to pop up on the screen. And it's all based on a secret key and a counter that is in place. And this message that pops up is the one that we're going to use as our one-time password.

We commonly see this used on token-based authentication, where you're carrying around different kinds of tokens. I have a token here, and that one-time password would pop up on my token generator. And that's what I would use to be able to authenticate. Every time you try to authenticate, every time you push the button on that token authenticator, it's going to give you a different hash every time.

There are both the hardware and software tokens that you can get to do this. So you're going to need some type of additional technology to make this work, either something physical that people can take along with them, or you'll need to install software on mobile devices that people carry with them.

**TOTP** stands for **Time-based One-Time Password**. In a time-based one-time password you're going to get a certain password based on whatever time of the day it happens to be. This is a little bit different than the HOTP we were just talking about, where you got a password based on a counter. Every time you use that password the counter would then increment. In the case of TOTP, it just depends on what time of day it happens to be. And that's what's going to synchronize these passwords together on both your side and on the resource.

For time-based passwords, obviously, time synchronization is very important. So you'll need to define a secret key and then time stamp and have everything synchronized via standard protocol such as Network Time Protocol. These timestamps usually increment every 30 seconds or so, although this value can be changed by the administrator. So you would put in your username and your password, and you would put in whatever the latest 30 second code happened to be.

And if you didn't get that right you may have to restart the process. And if 30 seconds of gone by you may have to input a different password because the time of the day is now different. This is a very common way to provide one-time passwords. If you're using a separate one-time password generator for Google, for Facebook, or for Microsoft then you're probably using TOTP.

**Tags:** [certification](#), [comptia](#), [hotp](#), [one-time password](#), [security](#), [totp](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **CHAP and PAP – CompTIA Security+ SY0-401: 5.2**

The CHAP and PAP authentication protocols have been a mainstay of network computing. In this video, you'll learn how PAP and CHAP operates over the network and some of the advantages and disadvantages of using these protocols for authentication.

When we log into a network, when we connect to a server, we need some way to authenticate ourselves. And so there's a series of protocols behind the scenes that make sure that our username and password, or any other information that we're providing to authenticate, is received properly and we're able to get access to the resources that we need.

One of the first types of methods of doing this was an authentication protocol called **PAP**. And that stands for **Password Authentication Protocol**, and extremely simple, extremely basic protocol. If you recall Kerberos in an earlier video, there was a lot of encryption. There was a lot of communication to certain servers. It was a multi-headed process that took place.

**PAP** is nothing like that. PAP is incredibly simple in the way that it works. Everything that's communicated, in fact, with PAP is completely in the clear. There's no encryption here whatsoever. If I was to type my user name, [? ajames, ?] and my password, [? aprofessor ?], all of that will be sent in the clear across the network to the other side, where it would be interpreted. And then I would be allowed access or not allowed access.

Now obviously on today's modern networks, that is a painfully insecure way of doing things. You do not want to send especially password information in the clear. So generally, we don't see PAP being used very often. But we are using other methods of authentication that fortunately are much more secure than this.

One of those more secure methods is one called **CHAP**. It stands for **Challenge Handshake Authentication Protocol**. This is an encrypted message that is sent across

the network. Microsoft also has their own version of this called **MS-CHAP**, so you may see it referred to that way if you use a lot of Microsoft operating systems.

This is a three way handshake. The link is established to a server that we would want to authenticate to. And that server sends us a challenge that says, if you want to get in here, you're going to have to prove that you are who you say you are. Our client, us, is going to respond back with not the password in the clear, but a hash of the password.

And we're going to send that hash across the network to the other side. And at that point, the server is going to compare what it received in the hash with what it has stored as the hash of your password. And if all of those things match up, then you're allowed access to that resource.

Even after you gain that initial access to the server, the CHAP process is not quite done yet. It will continue to ask for challenges and get responses back every so often at a pre-defined time. And your username and password is probably cached on your machine. So you generally never see this happening. But behind the scenes, it's going to check in every so often and make sure you are really the person who would you say you are.

If we were to look at this graphically, we would probably see a message to pop up on our screen that asked for an authorization. And we would put in our name and our password. And although we're typing in our password with what it really is, for instance, password111, we are not going to send that over the network.

The application that is asking you for this is going to hash that. And it's going to send this. So when the server asks you to authenticate, you may send the user name in the clear but you'll notice that your password is completely hashed when it goes to the other side. And that's the information that the server is going to use to make sure that that syncs up.

An authentication method that was created by Microsoft is one called **LANMAN**. You may see this referred to as **LAN Manager**. This is one that was created from Microsoft and 3Com. They were just getting into the world of network based operating systems. So this is well before **Windows NT**. This is well before any of our **Windows Vistas** or **Windows Sevens**.

This was a very, very early type of operating system. And it's one that still needed a way, of course, to authenticate. So Microsoft has their own challenge response system in **LANMAN**. And it was very similar to what we saw with CHAP. But it was a little bit different.

For instance, it was only uppercase **ASCII characters** that you could only have a maximum password size of 14 characters. And notice that if you had passwords over seven characters, it split those off at seven. So if you had an eight character password, what you really had was a seven character and another one character password that it would save.

So already, the information that's being stored is not perhaps the best one to have there from an encryption perspective. And the passwords are not salted either, which means that they're always going to look exactly the same every time. We're not adding a bit of randomness into the password process when we're sending it across the network.

So there's challenges there with keeping that secure. There needed to be different ways to look at handling this LAN Manager configuration. So Microsoft tweaked it just a little bit to try to make a few things more secure. The update that to make it more secure came with Windows NT. And this was updated to something called **NTLM, NT LAN Manager**.

This is what was used in early versions of Windows NT. The password is now Unicode, which means it has a lot more flexibility on the types of characters you can have in there. It could be up to 127 characters long. And it's stored as a **128-bit MD4 hash**, which is

generally a lot more secure than the smaller DES hashes that were being used in the LAN Manager configuration.

But even that wasn't good enough a new version called **NTLM** version two came out. This came out with **Windows NT Service Pack four**. And this added some additional security. We had a new password response. There was an **MD4 password hash**, the same as what we had with the **NTLM version one**.

And there was a hash of the username and server name combined with that in there. So now we had a little bit more information thrown in there to make this a little bit more encrypted as it went across the network. This means it's not going to be exactly the same every time. There's going to be some things that are going to be randomized when that information is hashed and sent across the network.

And there's also this variable length challenge sent that has a time stamp, some random data, some domain name information, a little bit more details in there, so that we could make that conversation a little bit more secure during the authentication process. There are a few vulnerabilities you should probably be aware of when dealing with NTLM version one, NTLM version two. And part of this was created because there were two versions of this NTLM authentication.

Therefore, we had to make sure that older systems could authenticate. And if an older system didn't know how to authenticate with NTLM version two, then it may be locked out of the system. So unfortunately, a number of Legacy systems kept not just the NTLM version two password, which was pretty secure, but it kept the older insecure NTLM version one password as well.

The challenge is that if somebody gained access to that NTLM version one database, they would be able to have a much easier way to decrypt and figure out what people's passwords were. We also had a problem with NTLM, where it was vulnerable to what we call a credential forwarding attack. What this essentially meant that we could use the credentials of one computer to gain access to another computer.

Now obviously, that's not what you want to have happen as well. Microsoft has introduced bug fixes and updates to their operating systems to avoid these types of situations. But it's something you should be aware of if you're using a number of these Legacy authentication systems.

**Tags:** [authentication](#), [certification](#), [chap](#), [comptia](#), [pap](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Single Sign-on – CompTIA Security+ SY0-401: 5.2**

Providing a secure, single sign-on process is the goal of most network and application administrators. In this video, you'll learn how single sign-on works and how it can be used to create a seamless authentication infrastructure for your user community.

If you're in a medium to large environment, you may have heard of this concept of single sign-on. And it just makes sense. If you have so many different resources that you need access to, it'd be a lot easier if we could just authenticate one time, and we would have access to everything that's out there.

There's many different ways to do this. It's obviously are relatively complex process behind the scenes to make this happen. But we've already talked about Kerberos which is one of the more popular ways to provide a single sign on process to the network.

There are also a number of third-party options available, as well. You really don't see this much in smaller environments just because of the complexities that you would need to set something up like this. And by the way, how many different things do you really have

to log into, how much time are you really spending having to do that in a smaller environment?

Obviously, the cloud itself, and as we're using more and more services that are in more and more places anywhere over the internet, this is becoming increasingly a little bit more of a problem even in smaller organizations. So it's still an important concept we need to keep in mind, regardless of what size organization you might be.

Kerberos is a very common way to do single sign-on. You've got ticket granting services. You're getting the ticket. You authenticate one time, and then you're done. Everything else really happens behind the scenes.

You don't have to worry too much about where those tickets are going and all of the things that are happening with encryption behind the scenes. It's invisible to you as an end user. And it's simple. You just put in your user-name and password, and you're finished.

Every time you connect to a printer, every time you need to map a drive, it doesn't ask you or reprompt you for usernames and passwords, unless, of course, you don't really have access into those resources. You might be asked for additional authentication to be able to do that.

Now this is just Kerberos-ish but, of course, you're going to have to make sure that everything you're doing is doing Kerberos. If you need to gain access to a system that's not Kerberos friendly— maybe you're logging into a service that you use across the web, maybe that web can't communicate back to your Kerberos servers to be able to authenticate. So maybe that doesn't work for everything that you're doing. So keep in mind that whatever system you're using for single sign-on is only going to work if everything plays together to be able to make that happen.

Software as a service has really changed the way that we use applications and obviously creates some complexities in dealing with single sign-on. So you've got these web services that might be on Amazon, it might be a web conferencing system, it might be a location where we're storing data to be able to recover that later.

Each one of those individual systems are different companies. They're different databases. It's a different login process. So to be able to consolidate and use those things with a single logon, and a single sign-on becomes a little bit more complicated. To help address this need that many organizations have to verify and reliably authenticate people into all of these different services automatically, there are number of third party companies and pieces of software that you can find out there.

One example of this is OneLogin. They have a catalog of over 1,500 applications that they can do single sign-on with. They can even use multi-factor authentication with the single sign-on process. If you want to read a little more about it, it's at onelogin.com. It's an interesting technology in itself.

As organizations have so many people that are using so many applications in so many different places, the single sign-on capability becomes even more important for the security of your organization.

**Tags:** certification, comptia, Kerberos, saas, security, single sign-on

**Category:** CompTIA Security+ SY0-401

## **Federation and Transitive Trust – CompTIA Security+ SY0-401: 5.2**

Our networks rarely operate in a vacuum, and it becomes increasingly important that we provide the proper security posture to other parts of our network and to third-parties. In this video, you'll learn how federation and transitive trust can give you more control over who goes out and who comes in.

The concept of federation allows you to provide access to people who may not necessarily be part of your organization. You might need to provide access to people outside of your company. These might be partners, or suppliers, or customers, but you may not necessarily want to provide the authentication yourself. The way that we do this, is by creating a Federated network.

Someone would authenticate and gain access to your resources based on authentication that would come from somewhere else. For instance, you can go to a website and instead of creating a new account on that website, there might be a link that says that you could log into this website by using your Facebook credentials. And there would be a process in place to allow you to authenticate to Facebook, but still gain access to the resources on this third party site. That's because a trust relationship has been created behind the scenes before you arrive between this third party site and Facebook. The degree of trust was also created because when you log in to the third party site, you may only allow that site to have certain rights and permissions to what you are doing with your Facebook credentials.

These trust relationships need to be put in place very early on when you're establishing relationships between these organizations. Once they are in place, it becomes very difficult to change them so you want to really plan this out there's something called a one way trust, where domain b would trust domain a, but the other way round doesn't work. Domain a does not trust domain b. You might also find two way trust, where both domains are equal peers with each other, and they both will trust each other equally.

Some of these trusts are non-transitive. We may create a trust from domain a to domain b, but we would not allow domain b to extend that trust to other domains. Or the trust may be transitive, where domain a trusts domain b, domain b trusts domain c. Therefore, domain a would then trust domain c. These trust relationships are extremely important. They need to be well planned out, and they can be a very powerful tool to allow access to the resources in your environment.

**Tags:** certification, comptia, federation, security, transitive trust

**Category:** CompTIA Security+ SY0-401

## Roles and Account Credentials – CompTIA Security+ SY0-401: 5.3

The user role can be a useful criteria for security management. In this video, you'll learn the best practices for role-based management and how shared accounts and credentials should be managed.

One of the challenges for a security professional is assigning the right roles of people to the correct resources. We have so many different resources on our networks. There are printers that people have certain permissions and access to. And certainly, files and folders that are on servers have many different permissions and rights depending on who needs access to those particular pieces.

We usually assign access to these resources based on someone's role. This is much easier than taking each individual person and assigning individual rights and permissions to all of those different resources. Instead, it's so much easier to say if you are in the marketing department then you have this kind of access to this particular printer, or this particular share on the network. This makes it much easier, so that when somebody's removed from the marketing department then they would automatically lose access to those particular resources.

We want to make these definitions to be very specific to someone's role. But we don't want to make it so specific that it becomes cumbersome to manage. So the marketing department might have some global roles associated with it. But you might need different roles for the management of the marketing department versus the rest of the marketing department. And how you segment out and decide what types of groups or types of roles you create is going to be based on the business needs of those particular areas.

If you've then assigned different levels of access to a resource— one for the marketing department and a different level of permissions for the marketing managers— then whatever system you're using, or the resource itself, has to be able to make a determination of what role to put you in. Do we put you in the marketing manager's role? Or do we put you in the more restrictive marketing role? And the agreement over whether somebody gets more permissions or less permissions is generally based on the type of role system and permissions that you are using in your environment.

A shared account is when more than one person knows how to authenticate in as a particular username. This is sometimes limitations based on the device that you're using, maybe it only allows for one particular username. But generally, most of the systems that we are using allow you to build separate account names so that you don't have the situation where you have shared accounts.

This is usually a best practice to avoid shared accounts, although you'll find some organizations that use the administrator account and is sharing that across multiple users. This is probably not the best idea. But it's something that is allowed by the operating system.

Just one of many reasons why this is a bad idea can be based on the idea of auditing. If something is changed or modified on a file server, or on the network, it would be nice to know who made that change and when they made that change. But if multiple people are logging in as administrator, for example, then all you know is that the administrator user made a particular change. And you have no idea exactly which user did this.

The concept of narrowing this down to a very specific person is called non-repudiation. We know that if we have a particular user's name pop up then it really was that user. But if everybody's sharing the same account there's no way to have non-repudiation.

Shared accounts are also much more prone to be compromised. When you have so many people that know the username and password for an account it's just that much more

likely the that password is going to get out. And if we do need to change the password on the account then we have to inform every person who is sharing that account name that the password has changed. And then we have to find a way to get them that new password in a way that is secure.

The idea of shared accounts is something that creates a lot of complexity on the security side. And the best practice is to avoid it at all costs.

We want to be able to protect these login credentials at all cost. It is this username and password that gains access inside of your network. And we want to make sure that only the authorized people get access to those particular resources.

One way to protect against somebody gaining access to this is to protect both the username and the password. At no time should you embed some of this information within the application itself. Sounds so easy, if you could just walk up to a machine and it would automatically login for you. But every time you're doing anything automatically with credentials that means that that user name and that password is being saved in some way inside of your system. If possible, we would like to store those only in your brain, and not in any type of automated system.

We also want to be sure that if we're sending that authentication information across the network that it's also protected and encrypted. Very early on in the world of computing we were sending information across the network in the clear. But we found out very quickly that that is an easy way to have your credentials stolen.

So if you are going to send a password over the network, it's always best to send a hash, or some encrypted form of that, to be sure that no one would be able to SNIP the network, capture those packets, and then have access to your login credentials.

**Tags:** certification, comptia, credentials, management shared account, role-based, security

**Category:** CompTIA Security+ SY0-401

### **Group Policy – CompTIA Security+ SY0-401: 5.3**

Microsoft Windows provides some powerful management tools to help security everyone who is connected to a Windows Domain. In this video, you'll learn about Group Policy and how security managers can use Group Policy to help tighten down their security posture.

If you're administering a lot of different systems, then you've probably run into the scenario where you've needed to manage or change one particular feature or setting on many different devices. And instead of going from one device to another to another individually, in the Windows world we do this with something called group policy.

This group policy management allows you to select different capabilities of the system and be able to manage or set those across entire groups or even your entire network and every computer within it. There are literally thousands of configuration settings within Windows, and group policy allows you to easily administer that from one separate application. This is something that's a little bit different than setting permissions to an NTFS folder or a share that's on the network. Those are very specific to gaining access to data.

The Group Policy settings allow you to change how the system is configured. Here are some good examples of user rights assignments within group policy. If you wanted to allow or not allow someone to change the system time, you can do that within group policy. You can allow or not allow someone to change the time zone. You can adjust memory quotas for a process or allow or disallow someone from logging on locally.

As I mentioned, there are thousands of these that you can choose from, and that gives the administrator of these systems a lot of control. This is a capability that is generally linked to Active Directory. When you have devices that are authenticated to one central directory system in Windows, you can then manage all of those devices. And you can even break out these group policies by different areas of the company or even different groups. If you wanted to set a certain set of group policies for the marketing department and have a different set for shipping and receiving, you can do all of that from the Group Policy Management editor.

There are generally two different areas in group policy that we would look at. One is the administrative policies and the other one is the security policies. In the administrative policies, we would do things like add or remove programs. Allow people to change sounds or prohibit them from changing any of the sounds. Allow or disallow font downloads. These are settings that you can really tweak and modify to get exactly the user experience that you would like, and to make sure that the desktop environments that they're using are able to work without any type of problems.

The security policies are obviously much more focused on the security side of the operating system. We can specify what a minimum password length might be. We could require that someone authenticate to a system and must use a smart card during that authentication process. Or you can do things like enforce certain log in restrictions on the user. As you can see, you can spend a lot of time working on these group policies. But you're also able to create a desktop that's going to provide accessibility for your users, and at the same time keep everything secure.

**Tags:** certification, comptia, group policy, security, windows

**Category:** CompTIA Security+ SY0-401

### **Managing Password Policies – CompTIA Security+ SY0-401: 5.3**

How secure are your passwords? In this video, you'll learn about password complexity, length, expiration dates, recovery processes, and account lockouts.

When you're deciding on what password policies you're going to use on your network there are a number of best practices you should keep in mind. We want our passwords to be strong. We want them to be very difficult to guess. So we want to use a password that, perhaps, is more than just a single word.

We want this to, maybe, be a passphrase that uses multiple words. And make sure that we don't use obvious passwords. You don't want to use the name of your children, or the name of your pets. Maybe you'd like to also mix the upper and lower case of the letters, even if it is a single word. That way it would be a little bit more difficult to guess or to brute force. And we want to, perhaps, even use special characters.

We don't want to replace things that are very common. For instance, you don't want to replace an O with a 0. The bad guys already know that particular trick. Instead you want to embed special characters within the password or the passphrase that you're using.

We consider strong passwords to be eight characters or longer. Once you get up into those longer number of characters it becomes much more difficult to brute force just because of all of the different options that have to be used for every single character. If someone has to check for uppercase, lowercase, and special characters across all eight of those characters it becomes much more difficult to go through every possible permutation.

We also want to be sure that the same passwords are not reused over and over again. Generally the systems that we're using can remember all of the old passwords you were using. And as the security administrator you can generally tell the system how many of

those passwords to remember. That way if somebody does gain access to one of your old passwords they would never be able to use it because you would never be reusing that password again.

If someone does gain access to your account we want to limit how long they'll have access with that particular password. So what we'll require is that our users change their password after a certain amount of time. Generally this is every 30 days, or 60 days, or 90 days. If you're in an environment that's very secure it may even be shorter amounts of time so that if somebody does gain access or compromises in account we are going to limit the amount of time they have access to those systems.

If someone forgets their password and needs their password reset we want to be sure that that process isn't something that's easily done. We don't need the bad guys from the outside calling in and saying, hi, I'm Professor Messer. I've forgotten my account. I need you to reset my password, and then simply have somebody on the phone able to do that. There needs to be a verification process, or perhaps a secure method to communicate directly with the end user to make sure they understand what changes are being made to their account.

Many operating systems will automatically disable an account if somebody tries to log in with the incorrect credentials a certain number of times. This is usually what we would like to have happen in these operating systems because we don't want somebody to be able to keep trying over, and over, and over hundreds or thousands of times until they will finally figure out your password and gain access to your account.

This could be a big problem for service accounts, however, if somebody's trying to log in as a service account and you have a lot of automated services running with those credentials, and then that account gets locked out, it could then cause those services to no longer work. So there's usually a balancing act between creating an account and having that lockout time for your service accounts.

When someone leaves your organization you may be tempted to simply delete their account and therefore that would keep them out of logging into systems that they previously had access to. But the best practice is actually to simply disable their account. This still prevents them from logging in, but it retains all of their files and all of their settings. And if you need access to any of those things, now that they left the organization, you can easily get it by gaining access to their systems behind the scenes.

This is a good best practice not only for small, but also large organizations. And there's usually a process in place to finally remove all of the information they were working on, and move it to a separate account. And at that point you can then delete the older accounts.

**Tags:** certification, complexity, comptia, expiration, length, lockout, password, recovery, security

**Category:** CompTIA Security+ SY0-401

### **Privileges – CompTIA Security+ SY0-401: 5.3**

User rights and permissions can be complex to manage. In this video, you'll learn about user management, group management, and role-based management techniques.

In many ways getting someone authenticated to the network is the easy part. Once they get here, we need to make sure they have the right privileges to access the resources that they need to be able to do their job. And it's a challenge to keep track of all of these things.

We have to figure out what rights a user might have to a folder, to a file. We need to make sure maybe they only have read access to a certain part of the network, but read and write access to another, and there's all kinds of overlapping policies associated with this.

So we don't know if there's operating system changes, things that are associated with their group permissions, their permissions associated with the user they might be using. The individual file might have rights and permissions associated with this, and all of those interact together. To be able to set some of these privilege we do a privilege management type for user management, group management, and role-based management.

User management is something that's very easy to do. It's done on a user by user basis, very simple, but it's also very unsophisticated. There's not a lot of flexibility that we have with that. We go to a specific user, we grant them specific rights, and you're done. You've now created the rights and permissions and privileges for that user. This is something we go to each individual user, and we carve out exactly what rights and what privileges they might have.

Obviously this becomes a little bit difficult to manage, especially if you need to make one tiny change, you have to go into each individual user account to make that change. So obviously, this is not something that's going to scale very well. If you have a large environment, you could be spending all day going into every single user account and changing every privilege for every person in the organization.

Group management is a little bit different. For group management we're setting privileges on what you as an individual are doing in the organization. We may put many different people into a group. Maybe we have an accounting group, a marketing group, a shipping and receiving group, and we set privileges for the entire group all at once. And If we need to change or modify those privileges, we're doing it for everyone who might be in that particular group.

If somebody needs the privileges, somebody joins the marketing department, we put them in the marketing group, and like magic they suddenly have access to everything the marketing group needs access to. So it makes a little bit more of that administrative process much more streamlined for us.

Now users can, of course, be members of multiple groups. You may be in the marketing group, but you might also be in the Florida group, or you might also be in the east coast group. There's three groups right there, and there may be different permissions for the exact same resources in each one of those groups.

So now you have to figure out what are those effective permissions. If you're in the marketing department, you need read and write access. If you're in the east coast group maybe you need read access. Which one takes priority? And what takes priority is really dependent on the operating system and the methods that you use for those operating systems to determine that.

So it's not quite as straightforward as you might think, but as long as you understand what's involved in trying to calculate or determine those effective permissions, this becomes a very nice way to be able to control a lot of different rights and permissions all at the same time.

Role-based management takes this idea one step further where we're really setting some very fine grain controls for what people do in the organization. So perhaps instead of having a big marketing group, maybe I have a field marketing group, maybe I have a technical marketing group. Those two people are in the marketing department, and they have the exact same rights and permissions for marketing, but maybe I need to break them out and have different controls for really their roles within the organization.

We also might want to think about how we're going to create these because you could end up creating a lot of different roles. Obviously a lot more administration associated with this, because you could have HR managers, you could have accounting analysts, you could have IT project managers. There's a lot of different ways to go with the role-based management infrastructure, but it makes it very easy if you need to move people in and out of different permissions and rights as their role changes.

And in some organizations you're required to move from role to role to role at different times of the year. So from an administrative perspective, we can simply add people to new roles, and now all of the different privileges move with that user as they move into the new role.

And of course, you need to keep in mind in this particular role-based management, you're only going to be a member of one specific role. You can't be a member of multiples. That's not the idea behind this. This is something where we're recreating very fine grain control, and therefore, you're only going to be part of one role at a time.

**Tags:** certification, comptia, group, permissions, privileges, rights, role-based, security, user

**Category:** CompTIA Security+ SY0-401

### **User Access Reviews and Monitoring – CompTIA Security+ SY0-401: 5.3**

It's important to constantly validate your security posture. In this video, you'll learn how access reviews and auditing can provide you with constant feedback about your security implementation.

How's the security in your environment? Do you have the right permissions and settings so that people can access the files they need? And are you restricting the bad guys from accessing resources inside of your network?

The only way to know for sure is to perform some type of access review. This allows you to get into your systems and really understand how the security of those is performing across the board. You may be able to find misconfigurations or things that might have changed in your policies by simply looking over an auditing of what's happening on your network today.

This auditing should occur relatively often, because you need to understand what changes might be happening on your network. You should, of course, look and see who's been added to particular groups. You want to be sure a person who does not need access

to an administrator group is not a member accidentally of that administrator group. You'd like to be able to review the access control lists and make sure that people have the correct access to resources on your network.

You also want to find any accounts that you might have configured but ultimately had nobody use them. Sometimes this can be done with normal login accounts to, for instance, your Windows domain. Occasionally, it might be a third party system, like a VPN system that you set up originally for someone, and then they never ended up using that VPN.

In those particular cases you have a potential security risk by having that account there. And if somebody does gain access to the proper credentials, they could then access the VPN through that available account. It makes much more sense to simply disable that account, so that nobody can use it. And, of course, if there any accounts that are unnecessary on your network, you want to be sure that those are completely disabled as well.

It seems like going through all of these different steps would take a lot of time. And if you did it manually, it certainly would. There's many tools out there that you could use that go through not only these particular auditing points but many others as well. And they'll identify any red flags for you so that you can then go back in and do some additional research to see if this is really a problem and determine what you can do about it.

This auditing process will very often look through all of the different event logs that have been created on your infrastructure equipment, in your file servers, and anywhere else on your network. These logs usually have a lot of information inside of them. And they're usually specialize logs for application usage. There might be security logs that tell you when someone logged into the network and logged out and security types events. And, of course, there's audit logs as well, so that you can keep track of who made a change, at what date and what time and be able to backtrack and understand exactly what has changed on your network or in your devices over time.

All of these event logs together allow us to create an audit trail. And we can really track to see what has happened in the past. One downside of storing all this information is that it takes a lot of room. In even a mid-size organization, you could have terabytes and terabytes of audit logs and event logs to go through.

You don't really want to turn these off. There's a valuable amount of information that's there. And all too often, people will decide that that's just too much space to have on their desk. And they'll simply disable this completely.

It might make more sense to modify how much the log will save, so that at least you can go back a little bit in time and get some visibility into what's going on. Ideally, maybe you should get more disk space to be able to keep much more log information in your environment.

Having all of these event logs allow us also to determine if particular resources may have been accessed improperly. This gives us a specific date and time when somebody gained access to resource. And we can also determine exactly the process they went through to get that access. This now allows us, as security professionals, to not only understand what type of risks we were at when that occurred, but now we can put the proper motions in place to prevent those from happening again.

**Tags:** audits, certification, comptia, log files, monitoring, security, user access review

**Category:** CompTIA Security+ SY0-401

## **Cryptography Overview – CompTIA Security+ SY0-401: 6.1**

Our modern applications make extensive use of cryptography. In this video, you'll learn the basics of cryptography and some of the history of ciphers and secrecy

**Cryptography** is obviously extremely important in what we are doing today, with our networks and our computers. The term cryptography comes from the Greek word "**cryptos**," which means hidden or secret. And that's exactly, obviously, what we're doing when we're talking about cryptography.

There are number of features that cryptography brings to the table. One is confidentiality. This is the one we normally think about.

If we're going to send our credit card information across the internet we want to be sure that nobody in between would be able to see that credit card information. They'll have all of that information absolutely secret. Nobody's able to see it except for me and the person that I'm sending it to. Everything else, completely confidential.

The authentication and access control is also a capability that's provided to us through cryptography. This authentication method means that if I'm putting in a username and then a password, the cryptography around that password and the way that we keep it secret validates that it is me. I'm able to send that message in an encrypted form, and the other side is able to validate that that is indeed exactly the person I would be expecting there.

There's also a capability called **non-repudiation**, which means if I receive a message from you there's no way you can say that you did not send this. I can really tell that this was you that sent it. There's no way you can deny what you put into this message because I have a way to determine that it really did come from you, and that you really did write these things.

And that also brings us to integrity. We know that when we receive this message, and we check it with these cryptography and the methods that we're going to talk about, that we can be assured that nobody has tampered with anything inside of this message. If anybody was to change one word, or one letter, we would be able to notice it. And cryptography makes sure that when we receive a message we can be sure that it was not tampered somewhere along the way.

There are a number of common terms you'll need to know about cryptography. The first is plaintext. Before you encrypt anything it is plaintext, or something that we sometimes will call in the clear. If we were to send this information out over the network anyone would be able to read it. We haven't done any type the hiding or encrypting of this message.

Once we do encrypt the information it becomes **ciphertext**. We have taken our plaintext, we have applied a cipher to it, and the resulting encrypted information is the ciphertext. That **cipher** is what we're using to encrypt that message. It's an algorithm. It's a mathematical method that we are using to take the plaintext and encrypt it or convert it into something that people can't read. And that is the cipher that allows us to do that.

The art of cracking this encryption then is cryptanalysis. Having your cryptographers in a government facility trying to understand and make sure that the messages that they are sending our secure, and trying to crack the messages that are coming from other places, is a big, big business. And even though it's something that is relatively hush, hush and top secret, it's something that is happening every day.

There are many different ways to scramble up a message, and encrypt it in a way that no one would be able to read it. One common way is something called a substitution cipher.

This is one of the oldest methods that you'll find of encrypting data. This is also called Caesar's cipher because Caesar is one of the people that originated this method of sending information back and forth, and certainly made it very popular.

What he did was take a normal alphabet and he changed all of the letters down a certain number. In fact, a very common one is ROT13. You've rotated this group 13 steps to the right so that instead of writing a letter A, you would write the letter N. Instead of writing the letter B, you would write the letter O.

So your cipher is really this 13. As long as you know the number 13 you can take a message and encrypt it and decrypt it with that particular key.

This is obviously something that would turn this funny looking message into, hello world. It's a very simple one to do. And it made it very easy to send this message. It's encrypted, you can't read it. You would have to know exactly what you would need to do, how many methods, how many sections you needed to move this down and rotate it, to be able to read that message.

Another type of cipher is a transposition cipher, where we keep exactly the same letters but we just scramble them up and put them in a different format. And then you would provide the person on the other end with the key— what letter should go in what position. And they would transpose them back into the order that it was designed so that you could make hello world out of this scrambled bunch of letters that has exactly the right letters in it, they're just all in the wrong place.

We often see people trying to figure out these particular sections of ciphertext. And one of the methods it they'll use, especially on these older substitution ciphers and transposition ciphers, is to use frequency analysis. They'll examine the entire ciphertext and they'll try to determine how many times do I see the letter R? How many times do I see the letter Q? How many times do I see the letter W?

And they'll start creating a frequency table of how many times we're seeing different characters. And if we think that this is the English language, we know the letter E is certainly one of the most popular letters. T is the second most, A is the third most, O is the fourth most, and so on. And you can start substituting in some of these frequencies that we're seeing in our ciphertext to try to figure out exactly what the real message might be inside of that ciphertext.

Well obviously, in modern times, a **ROT13** or a **transposition cipher** is not really going to keep things very safe. So we started to see things like mechanical ciphers appear around the World War II time frame. This is one that Germany used. This is an Enigma machine, which took a message and encrypted it using a piece of machinery, which means you would have to have this piece of machinery on one end, and this piece of machinery on the other end, to be able to decrypt the messages that were being sent back and forth.

Being able to hack or crack that particular encryption code was an extremely important part of what happened during World War II. A lot of interesting history there.

There are also mathematical ciphers. And these are the ones that we will probably see the most of, especially in modern times, because if you can make it a very, very complex cipher then obviously it will be very, very, very difficult to be able to decrypt that message in some way. And we're going to go through a number of mathematical ciphers, not just in this video, but in many others were we will talk about hashing. We'll talk about doing symmetric encryption, and asymmetric encryption. If you're at all interested in getting into cryptography you will need to have an extremely strong background in mathematics.

We've talked a little bit so far about keys. And we've talked about if you know what the key is then when that encrypted message gets on the other side, you can apply a key to

it, with the correct cipher, and come up with the plaintext, back to the information we wanted to get our hands on. These keys can be very simple, like the ROT13 key. We needed that number 13. That was the important part of it.

If it's something like a **PGP or GPG**, and we'll talk more about those as well, the keys are very, very complex. This is my public key, for instance, that I use when I start encrypting information using some asymmetric encryption method. So you have to make sure you have the right kind of key for the message that you're getting so that you can encrypt it properly.

Another cryptography technique is a one-time pad. This is one where you would have a page of letters on one side. You have exactly the same page of letters on the other side. The person who is encrypting the message will go through each letter of the message, and they would combine the first letter of this pad with the first letter the message to come up with ciphertext. Then the second letter of the pad with the second letter of the plaintext message to come up with the next letter of the ciphertext, and so on.

They would go all the way through your plaintext message to create the ciphertext. And obviously this would be very, very difficult to be able to decrypt unless you had this exact key on both sides. You'd have to have that exact pad of paper, which is where the one-time pad comes from. You use that message one time, you get rid of it. And you have a completely new key, a completely new one-time pad, to be able to use. You use it one time and you're done.

So the next message, even if the first message was decrypted in some way, the second message someone would have to start all the way from the beginning again to try to decrypt that message. It is all of these different cryptography methods and cryptography features that really provide us with a very, very powerful way to keep all of our information private on our computers and across our network.

**Tags:** access

control, authentication, certification, cipher, ciphertext, comptia, confidentiality, cryptanal  
ysis, cryptography, enigma, frequency, integrity, keys, mathematical, mechanical, non-  
repudiation, pad, plaintext, security, substitution, transposition

**Category:** CompTIA Security+ SY0-401

### **Symmetric vs. Asymmetric Encryption – CompTIA Security+ SY0-401: 6.1**

Our encryption methods will generally use either symmetric encryption or asymmetric encryption, and sometimes both! In this video, you'll learn the advantages and disadvantages when using symmetric or asymmetric encryption types

In today's computer network environments we're using two major methods of encrypting data. We have symmetric encryption and asymmetric encryption.

Symmetric encryption is a method where we are using exactly the same key to encrypt information and decrypt the information. They're both using exactly the same information. So we know that if we need to encrypt data, and somebody on the other side needs to decrypt it, we need to somehow get that key to them so that they will be able to decrypt that data.

And because it is exactly the same information on both sides, it's the same key on both sides, you have to keep it secret. If somebody was to get that key somewhere in the middle, they would be able to look at all of the information that we had encrypted. They'd be able to see everything. And that means if the key gets out, or if it's lost or if it's stolen, you're going to need to create another secret key and get that secret key to the person who will be receiving the information that we're sending out in encrypted form.

So obviously this doesn't scale very well. If you give a secret key to one person, you might think that that key's going to be relatively safe. What if you give that secret key to 100 people? Is that key still going to remain safe? And since you have to have that key to be able to decrypt the information, anybody can get their hands on it and look at information. Now we've got a bit of a security challenge ahead of us.

Even so, we are still using symmetric encryption in many ways today, and that's because symmetric encryption is so fast to use. It uses so few resources when you compare it to asymmetric encryption. Because of that, you'll often see these combined. You'll see asymmetric encryption combined with symmetric encryption, to be able to have not only a secure environment where data is protected, but also have one that works very, very quickly and very, very efficiently.

The other type of encryption that you'll commonly see is asymmetric encryption. You'll also hear this referred to as public key cryptography, and you'll understand why in just a moment. This type of encryption method has really only been around since the 1970s. So in the world of encryption it's a relatively new capability, and it has allowed us to do quite a number of things in our technologies.

There are two keys needed in asymmetric encryption. If you recall, in symmetric encryption you had exactly the same key. But in asymmetric encryption, as the name implies, you have two keys. You have a private key, and this private key is something as also this name implies you, want to keep private. Nobody gets their hands on the private key except for you. You would not share this private key with someone else. You would not give it to someone else. Nobody else needs this private key.

The key that you're going to give to everyone else is one called a public key. Give it to everybody. Put it on a public server. Post it on your website. Stick it on your Facebook page. Everybody should have access to this public key. Nobody but you should have access to the private key, and that's because the public key is one that allows people to send you information in an encrypted form, but the private key is the only key that can decrypt that data.

And that makes this a very interesting method of storing and encrypting information, because if somebody was to encrypt data with your public key, they would not be able to decrypt it. Nobody would be able to decrypt it. Even if somebody got their hands on it somewhere along the way, the only way to decrypt it is with that private key, and that's why it is so important to keep that key private.

Not only is it private, we usually will put a pass phrase associated with it. We make sure that that key is very, very secure and nobody gets their hands on it. And in that way you can be relatively public with the information that you're sending. You don't have to worry about it so much because nobody could take that encrypted data and use your public key in some way to decrypt it. It doesn't work that way. You have to have the private key to decrypt that information.

So you can see, using this asymmetric encryption really gives us some new ways of encrypting data, keeping that data safe, and making sure that nobody can get their hands on that information. And when you combine asymmetric encryption with the symmetric encryption, you really have a lot of flexibility with how you're going to encrypt the data, send it to someone else, and be able to decrypt it on the other side.

**Tags:** asymmetric, certification, comptia, encryption, private \_\_\_\_\_ key, public-key, secret key, security, symmetric

**Category:** CompTIA Security+ SY0-401

### **Public Keys and Private Keys – CompTIA Security+ SY0-401: 6.1**

Asymmetric encryption uses two different keys to provide a secure channel. In this video, you'll learn how public keys and private keys can be used to encrypt data, create digital signatures, and create a secure symmetric key.

**Asymmetric encryption** is a foundation of this public key cryptography methodology. It's one where we have a private key and a public key. And we've talked a lot about these keys. But let's look at how they're made.

Whenever we're creating a key, it may be a single person we're creating this for, we're using a lot of mathematics to create this relationship between a public key and a private key. So we're building them at exactly the same time. We're using a lot of randomization. We're using prime numbers. And a lot of math goes into this to create a key that is a public key that we could give to anybody in the world and a private key that we would keep private to ourselves.

There's nothing that, if you were to look at them, looks the same about them. You would not be able to discern the public key, if you had the private key, and vice versa. And that is one of the things that makes this so powerful is that there is this mathematical relationship between them. But from the outside, they look very, very different. And it becomes very, very difficult to understand what might be encrypted unless you have the private key.

This relationship between the public key and the private key allows us to do some interesting things with the digital signatures. Digital signatures are ways to confirm that information has gotten from point a to point b without anything changing. There's non-repudiation associated with that.

The way this works is that, let's say, it's Alice is creating a document that says, I will pay \$500. And Alice is going to sign it with her private key. Obviously, nobody has access to that private key except for Alice. So we can be assured that Alice's private key that signs this is something that's very unique to her.

Now Alice is going to send that to Bob. And that message obviously says I will pay \$500. Bob looks at the signature at the bottom of this message and grabs Alice's public key to verify that it really came from Alice.

Obviously, Bob doesn't have access to Alice's private key. But that's OK. Everybody has access to Alice's public key. It's on public key servers. It might be on Alice's website. It's something that you would like everyone to have that available to them.

So Bob's going to grab Alice's public key and decrypt or verify that that digital signature of this particular message matches perfectly. If anything had been changed anywhere in the middle, Bob would have known it because that would have not verified properly.

And that really allows us to do some very interesting things with non-repudiation. We can make sure that certain documents that we're sending across the network are not changed. They remain intact. And we can start combining this encryption process with the digital signature process to make sure that the integrity of the files we're sending match when they get to the other side.

This relationship between the public keys and the private keys also allows us to do some interesting things with the math that allows us to build some symmetric keys from this, some that can be done automatically behind the scenes without sending these symmetric keys across the network. The way this works is, for instance, Alice certainly has access to her private key. She'll grab Bob's public key, combine them together, and create, through an algorithm, a shared secret key. This is a symmetric key that then Alice could use to encrypt other information and send it to Bob.

On the other side, Bob does the exact same thing. But, of course, he doesn't have access to Alice's private key. He does have access to his private key.

So he'll grab Alice's public key and his private key. It's the other side, it's the other pair of the same two that Alice were using, uses the same algorithm. And look, it magically creates exactly the same key between those two.

So when Bob receives this message that has been encrypted with this symmetric key, he simply does the same algorithm to come up with exactly the same symmetric key. And now he's able to decrypt the information on the other side. A very simple process to build that and one that you would not be able to do unless you had both this public and private key methodology.

**Tags:** asymmetric, certification, comptia, digital signature, encryption, key pair, private key, public-key, security, symmetric

**Category:** CompTIA Security+ SY0-401

### **Session Keys – CompTIA Security+ SY0-401: 6.1**

To provide a secure channel, both sides of the conversation need to share the keys that will be used during the session. In this video, you'll learn about session keys and the different methods used to provide a secure exchange of session keys.

We know that if we'd like to protect data that we send across the network, that we're going to want to encrypt that information. There's two major ways of encrypting data. And there are advantages and disadvantages of each of those.

With asymmetric encryption there are actually two keys. There's a private key that nobody gets to see. And there's a public key that you could send to every one, that everyone in the world would have access to. You can share this public key and not have to worry about anybody gaining access to it, because you can't decrypt information with the public key. You can only decrypt information with the private key.

This process of public and private key encryption, though, requires some additional resources. There's a lot more CPU cycles that go into encrypting and decrypting this information. So although it is a very secure method of transferring information across the network, there is some overhead involved in being able to encrypt and decrypt the data.

That's why, normally, whenever we're sending information in real time we'll use something like symmetric encryption. With symmetric encryption we use exactly the same key to encrypt and decrypt the data. In that particular case, we still need to share the information across the network on both sides. But if we need to send a symmetric key across the network then we'll want to also add additional protection so that nobody can gain access to that key.

This is a relatively fast way to both encrypt and decrypt data. So it's very common to use this whenever we're sending information across the network in real time.

To be able to encrypt and decrypt this data we need to share the keys on both sides of the conversation. And we can either do this in and out-of-band key exchange or an in-

band key exchange. An out-of-band key exchange means that we're not sending the key over the network.

We would be doing this over a telephone. We would send a courier with the key on a piece of media, or written on a piece of paper. Or we may exchange the key in person with somebody and let them know that we're going to use this key later to encrypt and decrypt information.

A much more common way to exchange keys is over the network with an in-band key exchange. Since you're sending this over the network, you have to add additional protection to these keys so that if somebody does gain access to these packets they would not have the information they needed to be able to decrypt our private data. We very commonly used in-band key exchange when we're sending a symmetric key. But we will use asymmetric encryption to encrypt the symmetric key and then send it across the network. Let's talk a little bit more about how that works.

It's very common to be this kind of fast security when we're working on a website. We'd like to purchase something with a credit card, we want to be sure that entire transaction is encrypted. That particular functionality is something that's handled via **SSL or TLS**. And we use this very fast encryption method using symmetric encryption to make that happen.

But before we can do this symmetric encryption we have to at least get that symmetric key across the network. And we have to do it in a way that ensures that the symmetric key can't be read by anyone else. The best way to do this is to create a symmetric key that you then encrypt and send across the network. You would combine both symmetric and asymmetric encryption to make this happen.

The first step is that your client machine is going to create a **symmetric encryption key**. It's then going to encrypt the symmetric encryption key using **asymmetric encryption**, and using the server's public key to do that. That means when the server receives this information it's going to decrypt it with its private key. And inside of that package, of course, is going to be that symmetric key that both sides will then be using to encrypt the data that's used on both sides.

This is the session key that's going to be in use. And as I mentioned, once that session is over, that key is thrown out. And if you need to perform another encrypted session a completely different key will be created, and the process happens all over again.

You also want to be sure that these keys are randomized and unpredictable. You don't want to use a session key that someone can easily determine what the next one might be, because that would gain access to all of your encrypted traffic. Instead you want to completely randomize the keys so that every session is using a very different key than the one before.

**Tags:** asymmetric, certification, comptia, decryption, encryption, in-band, key exchange, out-of-band, security, symmetric

**Category:** CompTIA Security+ SY0-401

### **Block vs. Stream Ciphers – CompTIA Security+ SY0-401: 6.1**

Some data transfer methods will encrypt data one byte at a time or in groups of larger data blocks. In this video, you'll learn how block ciphers combine data encryption with speed and efficiency.

When you've got a plain text bit of information and you're trying to encrypt it and get it into that ciphertext, there are a number of different ways to go through the actual encryption process. And when we're talking about symmetric encryption, one of those methods is

something called a block cipher. As the name implies, a block cipher is taking a fixed group of information and encrypting that fixed block all at one time.

And usually, these blocks are **64 bits long**. They're **128 bits long**. They're a size that is predetermined so that you're able to keep it the same when you're scripting and decrypting. And in this block cipher, sometimes you'll have some text that doesn't quite fill up the block. So occasionally, you will pad that data. Because you really do need a full block of 64 bits, 128 bits, whatever that size is, to be able to do that encryption.

There are a couple of things we should keep in mind with working with some of these ciphers. One of these is something called **confusion**. That means that the resulting ciphertext that you get should look very different than your key. There should not be a way that you could look at the encrypted data and somehow figure out the key based on what you're seeing inside of the encrypted data. That is called **confusion**. That means that it's a very complex and very complicated relationship between the key and the ciphertext that is created.

Another concept is one called **diffusion**. That means that your output should be very, very different than your input. And if you were to change just one letter of your input, your output should be dramatically different. You shouldn't have minor changes to the output when you make minor changes to the input.

That **diffusion** means that at least 50% of the output changes if you were to change one thing inside of your input. That way, you can always be assured that the output that you get, that ciphertext, is going to be as complex and difficult to figure out as possible.

Another type of cipher is called a stream cipher. And just as the block cipher was only used with symmetric encryption, stream ciphers are only used with symmetric encryption. The encryption is done, instead of entire blocks at a time, in a stream, it's done one bit or one byte at a time. It's all being done as the data is streaming by.

And this is something that can run very, very quickly. It usually uses very low hardware complexity. You don't need a lot of fancy chipsets, a lot of calculations to make this happen. So it's something that can be done very, very, very quickly as the data is streaming by. And that makes sense. If you're streaming data by very quickly, you would need to use a method that's able to keep up with that stream.

One important aspect to stream ciphers is that something called the **initialization vector**, the IV, should never be the same when you're starting to do some of the streams. Otherwise, somebody may be able to figure out that initialization vector that you're using and the cipher and the key that's being used and apply it every time you send data across the network.

So one very important part of cryptography and the way that people use stream ciphers and be able to use initialization vectors in general is to make sure your IV is always changing whenever you're using it to encrypt information.

**Tags:** certification, cipher, comptia, confusion, cryptography, diffusion, encryption, security, symmetric

**Category:** CompTIA Security+ SY0-401

## Transport Encryption – CompTIA Security+ SY0-401: 6.1

**Encrypting files** is important, but our network communication also requires additional privacy. In this video, you'll learn about the cryptographic techniques used to secure our network connections.

We are sending a lot of personal and a lot of very private information across the network. Sometimes we're sending credit card numbers, we're sending health care information. We're sending things we don't want other people to be able to see. So we of course, have put some cryptography into our network communication, into that transport stream so that we're able to make sure nothing going across the network is able to be seen by someone else. This can be seen in website commerce, credit cards, even emails that you may send back and forth.

There are plenty of places where we want to keep our private data very, very private. And to be able to do that, we need to use some of these cryptographic methods that we've mentioned already, and some that we're going to show you in other videos as well. These are using complex mathematics along with some very interesting techniques in being able to send information back and forth. And that's the real challenge— how do you send network information, sending crypted data, send keys and other information, without somebody being able to see that as it's going by?

Capture that data, and some way be able to decrypt that information. One very common way of seeing this transport encryption in use is when we're using a **VPN**, a **virtual private network**. This is one where you might have a laptop. You are away from the office. You're in a coffee shop, you're at your home office. But you still need to communicate with the resources that are back in your headquarters building, but you don't want other people to be able to see this information you're sending back and forth.

So you pop up some software on your desktop that creates an encrypted tunnel back to a **VPN concentrator** at your home office. So everything between your laptop and this concentrator is now in this private, encrypted tunnel, this virtual private network. This transport communication goes back to the concentrator. The concentrator decrypts that information and hands it off to the internal network so that all of the other resources internally will be able to understand what you're sending.

And if they're sending information back, then the concentrator's going to encrypt that information, send it through that encrypted tunnel. When it appears on your side, the software that you started is going to decrypt that information so that your workstation will be able to use it. Another common transport encryption method is the one that we use with our browsers, talking to web servers. We have both of those technologies able to perform encryption between each other. So let's look at an example of that.

Here's Facebook, obviously. I'm in Chrome. So I'm just at facebook.com. And if I click my globe, it even tells me that the connection to facebook.com is not encrypted. And I would like to show the difference between an encrypted method and a not encrypted method. This is a in the clear, non-encrypted method communication to Facebook. And what I want to do is start to capture with my Wireshark here. I've already set up an IP address filter here for the Facebook server that we're communicating with.

And I'm going to have it start this. All of these parameters look good. Let's just start a communication back and forth. We'll have that data go back and forth to that particular IP address. Now let's start up and just refresh this Facebook page so that some traffic can go over our network. And I'll stop this capture. And let's go all the way back up to the top and just look at the very first parts of this particular Comp communication.

There is some **TCP** communication starting up. And then we finally have some **HTTP** data going back and forth. And I want to be able, for instance, to see what is being sent. For instance, here is the **HTTP** data right here in the clear. You could even see the type of host I'm going to. You can see the user agent I'm using. It is Chrome. In fact, tells me right there. Tells the type of decoding and decrypting that I can do.

I can see cookie information inside of this. This is all completely in the clear. This is not what you would want to do if you were at a coffee shop. Let's do exactly the same thing now, but let's do it with HTTPS, or the encrypted form of HTTP. Let's start up a new capture file. And let's go back to Facebook. But at the beginning, right here of this Facebook page, <https://facebook.com>, and have it load the page. Now, it's loaded the same page that's here.

You can even see when we start looking at this that this identity has been verified, the encryption is in place with **128 bits**. This was encrypted data. So what does it look like now on the website? Let's have a look at the data that we sent back and forth. I'll stop this capture. If we flip back and look at our packet capture then, of this encrypted data, I've saved us some time and added a new filter here for the IP address of the encrypted server at Facebook. And if we drill down into just one of the packets, you can see that all of the information inside the packet is all scrambled up.

There's no words in there, there's no cookie in there that I can read. I don't know what browser someone might be using. In fact, I know nothing about this communication going back and forth other than it's HTTP type data. It's data that is all encrypted. So I can feel pretty sure by using these transport encryption mechanisms that if I encrypt some data on this end and I send it to the other side, I feel pretty good that nobody's going to be able to tap into that connection and see what's going on between our two stations.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [encryption](#), [security](#), [transport](#), [vpn](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Non-Repudiation – CompTIA Security+ SY0-401: 6.1**

Our encryption technologies not only provide a secure channel of communication, but they can also prove that the data we're receiving has not been changed between endpoints. In this video, you'll learn about non-repudiation and how digital signatures can ensure the integrity and authenticity of our data.

When we talk about non-repudiation we mean that information that we may have received can be absolutely attributed to somebody. There's no way that they can take it back. They absolutely said what it is that we have in this message right here.

When we talk about encryption, we talk about cryptography, we even add a different perspective to that. We add on a proof of integrity. We know the information that we received is intact. And we know that we can be assured that that information, when we received it, really came from the source. We can be absolutely authenticated with the person that sent this data.

And that may be difficult to do when we're talking about networks. We may not be able to see someone. We may not be able to talk to anybody. But by using these cryptographic techniques we can be assured that the package that we received really did come from the person that we think sent it.

We use digital signatures, in cryptography, to be able to check for non-repudiation. What we're doing is we're signing— we're digitally signing— a file. Or we're digitally signing a message. Just like you would sign a message at the bottom of a letter to prove that you're the one that wrote it, you are digitally signing a message at the bottom of your email, for instance.

And your email doesn't have to be encrypted. Your file does not have to be encrypted. In fact, you usually want to send your email in a non-encrypted form in some cases. All you're doing is putting a little bit of a digital signature at the bottom that people can verify that this message really came from you, and it was not changed between the time you wrote it and the time that I received it.

You're usually signing it with your private key then. Take your private key, sign the file or sign the message, and apply your signature that it outputs for you right there at the bottom of that message. The people who receive your message will use your public key, because obviously they don't have access, and you don't want them to have access to the private key. They're going to use your public key to verify that you're the one that signed it.

And that's the beauty of asymmetric encryption is that we're able to use our private key to do things privately on our side. And people can use our public key to be able to prove that this is really what we sent to them. And this allows us also validate signatures from other people. As long as I have everybody's public key, which is out there and available probably on a server or on their website, I should then be able to qualify and make sure that what I'm receiving absolutely came from them.

Let's try this out with some files on the internet, and see how this might work. Let's take an example of downloading a file that someone has signed for us. And we want to be sure that if we're downloading a file, and we've received a file, that it really did come from that person.

So here's an example one called the **Enigmail Project**. This is a plug-in that you can get to Mozilla applications, like Thunderbird, that will integrate the encryption methodologies that come from **PGP or GPG**. We'll talk more about those in another video.

But I've downloaded these files. I've downloaded both the file itself, that I'm going to use to install the program, and I've also downloaded a separate file that the people that originated this created, that contains the digital signature of that file. So you should be

able to see there's a digital signature that I've downloaded. And you should see some output that says that it's a good signature. And that means that the file that we received is the file that they signed to put on the web server.

So let's try that out. Here I am in my window to be able to look at my terminal screen. I'm going to show you the files I've downloaded. Here's Enigmail. This is the XPI. This is the file that would be installed.

And this is the signature that they created. So they took their private key, they said, private key, go sign this file. And it created this file right here. In fact, let's look at that. We'll look at the Enigmail 1.2-TB.

And here is the PGP signature. And here is the signature itself, right here— all this weird looking text going back and forth over it. And that's the end of the signature.

It's not very complex. That's the signature they just put at the bottom that said, my name is this, and I'm signing it at the bottom, and nobody else can prove that it came from me except me, because I put my signature on it.

So let's really see if the signature matches. We want to be sure the file we downloaded really did come from them. So I'm going to run my GPG program with a verify requirement.

I'm going to specify the signature they sent us, and say does this signature really— is it really signed? Is this file really signed with this signature of this person? And it's going to go out and see that the signature was made at this date and time, using this particular key ID.

It is a good signature from Patrick at masdev.org. It's key is not certify with a trusted signature. I've not had this key signed to prove that this really is this person. That's a very important part about this asymmetric encryption I just did, is it is a web of trust.

I would also have to be sure that the public key that I received really did come from Patrick. And so usually there's a method that we set up, we call them on the phone, we ask for their signature. We get a friend of theirs to prove that the signature public key that they have really is the public key from Patrick. There are number of different ways to do this.

But in this case, let's just assume that Patrick's key really is the one I have. And it really is obviously the one that signed this because I have a good signature for this file. And now I can feel very good about using this particular program because I know it came straight from Patrick. He signed it, and nobody's changed anything with that file anywhere in between.

**Tags:** authenticity, certification, comptia, cryptography, digital signature, encryption, integrity, non-repudiation, security

**Category:** CompTIA Security+ SY0-401

## **Hashing – CompTIA Security+ SY0-401: 6.1**

Hashing functions can provide us with some interesting ways of validating data. In this video, you'll learn the fundamentals of hashing and I'll demonstrate how you can validate a downloaded file by calculating a hash.

A **cryptographic hash** is a way to take some existing data— it might be a file. It might be a picture. It might be an email that you've created. And it's a way that you can create a message digest from it, which is really just a short string of text. If you were to look at a cryptographic hash— and we'll look at some in just a bit— you'll see there's just a bunch of letters and numbers put together, usually represented in a hexadecimal form.

Now what's nice about this is if you were to have a file and create a cryptographic hash from it, you could send that file to someone else. And you could say here's the cryptographic hash I created on my side. You do the same thing. And they'll take the file, create a hash from that file. If the hashes match up, then you know the file is exactly the same on both sides.

One very important characteristic of these hashes is that it's a one-way trip. I could not look at the hash, and somehow figure out what the original text was. That's a very important consideration with hashes. Because very often, we use these to take passwords and hash the password, and we store the password as the hash. We're not actually storing your actual password in plain text. This makes your storage of the password that much more secure. Because if somebody got their hands on this list of passwords or hashed passwords, they would have no idea what your original password was. That's an extremely important aspect of what we do with hashes.

But we can also use hashes for other things. For instance, they could be a digital signature. We know that if we were to take a file and create a cryptographic hash from it, we can give that file to someone else, have them also look at that hash, and make sure that what they've received is authentic, that nobody changed this file somewhere along the line. And they know that it came from you because you're matching up the keys that you're using to create that hash with— some very nice capabilities there. That means you don't have to encrypt the entire file that you're sending. You can simply send it along in plain text, but still be assured that nobody changed that file somewhere in between.

Let's take an example of this. I'm on the **GnuPG** webpage. This is the open source version of PGP— an absolutely free version. It's called **GNU Privacy Guard**. And I've downloaded a file. You can see in this download section that they have a number of files here. And along with the files, they have put the sha1 checksum for this. This is the hash of that particular file.

And I've downloaded this one right here— this GNU PG 1.4.11 tar gz, which is supposed to have this checksum. So let's see if it does. I'm going to flip over to my terminal screen, which is right here. Let's move it up so we can see all these things at the same time. Move this down.

Here's the file that I've downloaded right here. Let's see if that's what's really there. Yes that is. That's the GNU PG 1.4.11 tar gz that's on my hard drive. And I'm going to check it with that sha1 checksum, using openssl. And I'm just going to specify as a parameter sha1, and then the name of the file. And openssl will perform this sha1 checksum for me and create this hash from the file.

And ideally, that hash should match exactly what's on the website. And if you compare those two, you can see they are exactly the same. We have mathematically gone through that file, created a hash of it, and now we can be assured that the file that is on my hard drive right here matches the file that was put out there on the web server originally.

## **Key Escrow – CompTIA Security+ SY0-401: 6.1**

In large environments, your encryption keys may be held by a third-party to ensure that the encrypted data can always be recovered. In this video, you'll learn about key escrow and some of the business cases where key escrow should be used.

When we talk about escrow we're talking about a third party that's holding something for us. This may be money. It may be a document. In the case of cryptography it's usually a key. It's the encryption keys.

And the encryption key, for some third party to hold it, needs to be stored so that we would have some way to decrypt information should something happen to the original key. It's a very important part— especially for large organizations that are encrypting a lot of different things— you need some fail-safes in place. But you need to keep the key, obviously, somewhere very, very safe.

You don't want anybody getting their hands on that key. Because very often everything is built upon that key, whenever we're doing encryption. And we don't want people getting into our private information. If you are planning ahead and you are storing these keys, or you have a methodology in place to automatically store these keys, it can save you a lot of time and a lot of grief later.

Sometimes it's built into the process. Microsoft Windows, for instance, has methods to encrypt entire drives. And you can have those keys automatically stored as part of your Active Directory infrastructure. So depending on what you're using you may have a few options available to you for key escrow that are already built into your system.

For symmetric encryption, you're just keeping your key somewhere. Your encryption key and your decryption key are exactly the same. You obviously need to protect those at the end points. But having an extra one stuck in your safe somewhere, locked away, that nobody can get their hands on, would be handy as well.

You also want to think about what you're doing with asymmetric encryption. The public key, generally, is already distributed in many, many places. There may not necessarily be a need to keep a copy of your public key anywhere because it's just so accessible.

But the private key— which is the one that does the decrypting, it's the one that does the digital signatures— that's the important one that nobody should have their hands on. It makes sense to get an additional copy of that decryption key and have it already as part of something that you are escrowing or storing away.

Sometimes the process that we go through with this key escrow is just as important as having that private key itself. You have to think about what circumstances would arise that would require you to go into escrow and get that key? And who has access to the key? Is there more than one key?

Sometimes you can take a private key and split it up into smaller pieces so that you would have to have two or three people all come together in a room to put everything together to be able to have that additional decryption key be able to access that encrypted information. If you have the right process in place, and you have the right ideas behind what you're doing with key escrow, this can really be a valuable part of maintaining the integrity and security of your data.

**Tags:** asymmetric, certification, comptia, cryptography, encryption, key escrow, security, symmetric

## **Steganography – CompTIA Security+ SY0-401: 6.1**

When sending encrypted data, it's usually very clear that the information transfer has been obfuscated. In this video, you'll learn how steganography can be used to send information through covert, and sometimes unexpected, channels.

Steganography is a way to encrypt information or hide information. But you don't have that information right there in plain sight the whole time. It's derived from a Greek word that means concealed writing, and it is a way to secure things by making them obscure. Which in reality, isn't security. If you really know what to look for you can very often find these things. But if you just simply hide what you're doing inside of something else, makes it much more difficult to obviously see right in front of you.

The message is seemingly invisible, but it really is right there in front of you. In some cases, it's actually embedded within pictures, or embedded within sounds, or embedded within a document. So it may not be completely obvious to see with the human eye, but the message really is inside of that. What we're looking at is the cover text. This is the container for instance, like this graphical image of this network device's front slide. And what I've done is, you can see.

I've shown that there is some embedded information that shares IP addresses, device names, and Mac addresses that happen to be on my site. And they are embedded into this picture. They're part of this picture. Now, the picture looks like the normal picture for the network device's slide. But hidden inside of this graphic is that information. That's what makes steganography so interesting to me, is that this can be sitting right in front of us the entire time.

We would have no idea that it's really there. There are a number of ways that you can implement steganography in your environment. One way is to hide the data within the network packets themselves. Obviously, we can't really see the network packets. And they're going by so quickly, and there's so many of them. But if you were able to embed just one character inside of a TCP packet as it went from one device to the other, you could send many, many packets.

And it would be very, very difficult to see any of that data that we had hidden inside of that packet unless we knew exactly what to look for. Another way to do this, we're going to try this ourselves, is to use an image. We've already seen how we can take a picture and we can embed our own messages, and our own images, and our own documents inside of those pictures, a very unique way to hide information right in plain sight.

And here's a method right here on this screen where if you have a printer, well, especially an ink jet printer, you may have seen, if you look very closely, there are yellow dots that you can see. And a corner of the page that's printed on your laser printer, on your inkjet printers, on many printers these days, those dots altogether show the serial number, the make and model, and other information about the printer that you happen to be using. So if somebody happens to find, for instance, a document, and they want to know what printer created this document, they'll be able to see the exact serial number of your printer because it is steganography that is putting these tiny little dots here.

And now, we're able to track it back to your physical printer. I embedded some text inside of an image using this program, silent I. There's a version available for Linux, Windows, and Mac OS 10 that I'm using here. What I did was take an existing file— here's this image of this Network Devices— and it gives you the option of encoding information inside of it. Let me drag over the window so you can see this.

I can decide luminance, I can decide the jpeg quality, a pass-phrase even associated with that data if I want to assign one. And then I could put an entire message in here. I can put

entire file inside of this image. And when it is finished, it looks something like this. Here's one that I've already done this too. And this is the image. This looks exactly like the one we were just looking at. There it is.

You can see, if I go back and forth between them, there is a very subtle difference between them, but not much. And in fact, of the human eye, if you weren't really comparing it to this level, they'd look exactly the same. The difference, however, is I can click my decode option here. I can specify that this is a **BMP** encoding format. And I want to decode it. And now suddenly, all this information about the devices, **IP addresses**, **Mac addresses**, and everything on my network become decoded right from this image I have here.

Looking at the image, you would have never known all of this very important data was inside of it. But by using steganography, I'm able to hide it right there inside the image in plain sight.

**Tags:** certification, comptia, cryptography, encryption, security, steganography

**Category:** CompTIA Security+ SY0-401

## **Elliptic Curve and Quantum Cryptography – CompTIA Security+ SY0-401: 6.1**

The creation and use of cryptography has also included new ways to keep our data private. In this video, you'll learn about the use of elliptic curves to create encryption keys and how quantum cryptography can be used for spy-proof secure channels. An emerging technology in cartography is something called **elliptic curve cartography**, or **ECC**. **ECC** was created because of the constraints that we have associated with the calculations that we use in asymmetric encryption. As you recall with asymmetric encryption, we have to have a lot of mathematics that go into this. It takes up a lot of resources, a lot of CPU cycles. And that's because with asymmetric encryption, we have to calculate these very large integers that also happen to be prime numbers. And to be able to figure that out requires a whole lot of calculations.

Well, in steps elliptic curve cartography. Instead of using numbers to do this, let's use these curves that we've created. Every curve has a mathematical formula associated with it and a number of parameters that we can assign to that and each side can keep track of. At the end of the day, what this means is that this is much simpler to calculate. So it takes fewer storage requirements. It's easier to transmit, especially over mobile connections and mobile devices, that already don't have a lot of memory and a lot of CPU, can now use a simpler way of performing this asymmetric encryption.

Another emerging technology in cartography is quantum cartography. And like the name implies, this is using quantum physics and applying that into the calculations and methods of encryptions that we're doing inside of our cartography. What we're doing is this mathematical movement of these particles is now being used as part of our encryption mechanisms.

We are using some practical examples of this. There's something called **quantum key distribution, QKD**. And when you have two users that are sending information back and forth— for instance, they may be communicating a shared key between each other. And with that type of communication, you want to be sure that nobody in the middle is also able to see that shared key. It should be something that only the endpoints are aware of.

Well, with quantum key distribution, if any third party happens to try to look in there, it's very, very obvious that it has occurred, because with quantum physics you can tell when somebody is looking into the middle of what's going on. This is a very practical use. One that is being used today in cryptography fate. And we will certainly see elliptic curve and quantum cartography being used more and more as this particular kind of technology evolves.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [elliptic curve](#), [encryption](#), [quantum](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Perfect Forward Secrecy – CompTIA Security+ SY0-401: 6.1**

When a web server uses a single private key to encrypt all communication, then all communication can potentially be decrypted afterwards with this single bit of data. In this video, you'll learn how perfect forward secrecy can be used to create encryption keys dynamically for every encrypted session.

When you want to encrypt data sent from a browser to a web server, we commonly use **SSL**, or what's now called **TLS**, to perform that encryption. We're doing this by exchanging a session key between the client and the web server. And that session key is encrypted as it's sent over the network with a key pair— what's called an **RSA key pair**, which consists of a private key and a public key on the server.

Since every session key is encrypted with this public and private RSA key pair, if you somehow gain access to that RSA private key, you can then capture every bit of traffic across all sessions going to the web server. And then afterwards, you can decrypt all of the data made from every single session that was encrypted to that web server. This is obviously a single point of failure for every bit of website encryption. Once somebody gains access to the private key, they effectively also gain access to view all of your encrypted communications.

If you wanted to see what type of key pair was being used for this key exchange, you can look at the details. This is from my browser from JP Morgan Chase. And you can see the to that web server down at the bottom, it says my connection is encrypted with **128-bit** encryption. It's using **RC 4**, 128 with SHA-1 for message authentication and RSA as the key exchange mechanism. That tells us that RSA is what we're using for that public private key pair on that particular web server.

To get around this problem of having one private key that can then decrypt everything going to your web server, you want to use something like **Perfect Forward Secrecy** or **PFS**. This changes the way that the key exchange operates. It does not use the private **RSA key** that might be on your web server and instead uses elliptic curve or perhaps **Diffie-Hellman keys**. And these are ephemeral keys that are used just once and then never used again.

This means that even if somebody did collect all of the data going to your web server and they did have access to the private key of that RSA key pair, they would still not be able to decrypt this information, because every session is using a completely different private key to make this symmetric key exchange. As you might imagine, this Perfect Forward Secrecy process or PFS does require additional resources and computing power on your servers. That's why not every device out there is going to be using PFS, but we're starting to see more and more web servers take advantage of it.

Another reason people may not want to implement Perfect Forward Secrecy is that not every browser understands how to encrypt information to a web server using PFS. And if you're very concerned about having the largest number of people access your site securely, this may not be the right encryption method for you to use, at least not at this time. One website that is using this PFS method of transferring the keys is the Professor Messer website.

And if we look at the encryption details for my server, you can see that it says the connection is encrypted and authenticated using AES 128 GCM and uses elliptical curve Diffie-Hellman ephemeral RSA as the key Exchange mechanism. This means that it's using this PFS, or Perfect Forward Secrecy, to be able to have a different method of exchanging that symmetric key with every single session to the Professor Messer website.

## **WEP vs. WPA – CompTIA Security+ SY0-401: 6.2**

802.11 networks rely on encryption to ensure the security of all wireless traffic. In this video, you'll learn about WEP and WPA encryption and the dangers of using the wrong encryption on your wireless network.

Encryption over wireless technology becomes really important because wireless technology is radio waves. It makes it very easy for anybody to listen in on the right frequency, and really see what's going on on your network. So the solution has always been, if we're going to send information over these airwaves, let's make sure the data that we're sending back and forth is absolutely encrypted. And we'll make sure that everybody has the access information they need to be able to unlock, unencrypt, see what's happening inside of those data streams.

The idea is that only the people who have the password will be able to make any sense about what's going on. And we've applied two different kinds of encryption technologies through the years. One that's called **WEP** and the other one that's called **WPA**.

**W-E-P**, or **WEP**, was the **Wired Equivalent Privacy** that was introduced when 802.11 networking was introduced. And this technology uses two different levels of encryption, at the time. Depending on where you were in the world you could either have a **64-bit key** or a **128-bit key**.

Unfortunately, in 2001, some significant cryptography problems were found with the WEP protocol. What we found was that the first bytes of the output key stream are what they call strongly non-random. That means that the information at the beginning of this data that we were sending was something that we could easily tie back to the actual key.

And this would create a problem if somebody collected enough packets and put them through a process they could determine, with a relatively good percentage, what the key was for the wireless network. And they would then be able to access everything going back and forth over that network. In some cases, these days, it takes just a matter of minutes— sometimes even less— to be able to determine what a WEP key might be on a network. And because of that it is of course highly recommended that nobody ever use WEP.

When we found out that **WEP** was not going to be a good encryption method, we all scrambled to try to find out what we can replace it with. And what we came up with was **WPA**. That stands for **Wi-Fi Protected Access**.

This was **RC4**— which was the cipher we were using with the **WEP**— but it included a new **TKIP**, a **Temporal Key Integrity Protocol** mechanism. And it sent the initialization vector across the network as an encrypted hash, which was something that was not being done before. Every packet that goes across gets a unique encryption key. That was not the case with WEP. And this was, ideally, a short term workaround because we were able to perform WPA on the same hardware, for the most part, that WEP was running on.

Encryption methodologies obviously require overhead. There's calculations to be done there. This was a little bit of a heavier load on these access points. But it was something that was relatively compatible with the method we were doing previously.

But it was just a short term workaround. What we really needed was a more long term solution. And so, very quickly after WPA came out, we came out with WPA2. This began in 2004 for this certification.

The RC4 component was replaced with **Advanced Encryption Standard**. And there was something also added– **CCMP**– Counter Mode with Cipher Block Chaining message authentication code protocol. You can see why we call it **CCMP**. That particular component replaced **TKIP**. So we took the whole **RC4** and **TKIP** thing and replaced it was something better and stronger in the way of **AES** and **CCMP**.

You may also see, if you're configuring an access point, something called WPA2 Enterprise. In those cases, what we're referring to is in an enterprise you may not be giving out a key. What you may be doing is requiring people to authenticate via 802.1X.

So anytime you see the word Enterprise after the encryption type that is referring to something like a RADIUS server that might be in the back end, that that's providing the type of authentication for you, and applying a lot of those wireless configuration settings for you automatically.

**Tags:** [802.11](#), [certification](#), [comptia](#), [cryptography](#), [security](#), [wep](#), [wireless](#), [wpa](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Cryptographic Hash Functions – CompTIA Security+ SY0-401: 6.2**

There are many hash functions for many different circumstances. In this video, you'll learn about MD5, SHA, RIPEMD, and HMAC hash functions.

The **MD5 algorithm** was created by Ronald Rivest. He is one of the fathers of cryptography. He's been doing this for quite some time. If you have an opportunity to go out to YouTube and look at some of the presentations he's given, he really is one of the founders and brilliant thinkers in cryptography. The MD5 hash algorithm, itself, was published in April of 1992. As the name implies, MD5 comes after MD4. The **MD5** message digest algorithm is a **128-bit hash value**, so the information that you get once you hash something is **128 bits long**.

In 1996, however, there was a discovery of a number of collisions that were found with MD5, and one of the things that you'll notice as you examine hash algorithms is one of the biggest challenges they have is to make sure that there can't be two separate pieces of information that end up creating the exact same hash. That's called a collision, and in the world of hashing, that's a bad idea. When they found these in 1996, it was a pretty bad set of vulnerabilities, and they realized this particular algorithm is not very resistant to these types of collisions.

In fact, in December of 2008, researchers created a certificate authority certificate– this is a pretty big deal– that looked absolutely legitimate when you did an MD5 hash against that certificate. And they were able to build other certificates– these are the kinds that might be used on web servers, for instance– that appeared to be completely legitimate and issued by a third-party provider of certificates. So someone, technically, could take that certificate, put it on their web server, and your browser thinks that that certificate is absolutely valid, and that is a very, very bad idea. That means that I could pretend to be Microsoft. I could pretend to be eBay. I could pretend to be anyone.

To give you a feel for what these collisions look like, these are two separate pieces of information. Everything in red is text that is different between them, but everything else is exactly the same. But clearly, those are different pieces of information, and unfortunately, the MD5 comes up with exactly the same hash. And that's our collision right there. That's what we're trying to avoid. And right after this, turns out they ended up not using this

particular method to create these certificates any longer. Rapid SSL was decided not to really release or provide any of those types of certificates any longer because of these vulnerabilities that were found in MD5.

Another common hash algorithm is the **Secure Hash Algorithm** or **SHA**. Some people say **S-H-A**. It's one that was created in the United States by the National Security Agency, a government agency within the United States. It is also a Federal Information Processing Standard. So it's one when the government creates these standards, they decide to roll it out across all of the federal agencies, and that's the method that they use to provide certain hashes of their important information.

One that was widely used is **SHA-1**. This is a **160-bit digest**, so a little bit bigger than the MD5 we were just looking at. Unfortunately, again a common problem with hashing algorithms, in 2005, there was a publication that talked about collision attacks that could occur with SHA-1. So the natural progression, then, is to create one that's a little bit better, and **SHA-2** was released. This is now the preferred variant of this **SHA hash algorithm**.

This is a bigger digest, **512 bits**. The idea, usually, being that if it's a longer number of bits, it may be more difficult to find collisions between the different hashes. **SHA-1** is now retired for most US Government use. They've all been said, there's problems with using SHA-1, collisions are there. Everybody please start moving all of your different applications, all the development that you're doing, and all the products that you use to provide this hashing over to the more secure SHA-2 standard.

**RACE-MD** is an entire family of different hashing algorithms. It was created by **RACE**, and this **RIPEMD** stands for **RACE Integrity Primitives Evaluation Message Digest**. That's a mouthful. **RACE** stands for the **Research and Development in Advanced Communication Technologies in Europe**. So this is a European agency that was really created so that there could be some standards around communications through all the different countries in Europe. This is a centralized standard. There is centralized management associated with the technologies that they're creating. So this hashing algorithm, or sets of algorithms, was created just for this purpose.

The original version of this, the **RIPEMD** was found to have collision issues in 2004, and because of that, they've now moved to a **RIPEMD-160**, which, to this point, does not have any known collision issues associated with it. This is an interesting mix between **MD4** from a design perspective, but it has similar performance characteristics to **SHA-1**. So there's a nice balance there between the usability of this hash and the speed at which they're able to use it. There's also other standards out there, **RIPEMD-128**, **RIPEMD-256**, and **RIPEMD-320**, and obviously, the different hashes might be used for different things.

When you apply a hash algorithm to a file or a document or an email, you end up getting this nice little signature at the bottom. So all you really know is the document that you've received is exactly the same as the document that was sent, but you can't really verify who sent the document. So this little technique, which is the **Hash-based Message Authentication Code**, or **HMAC**, is one where you take a secret key and you combine it with the hashing process so that on the other side, you can apply the same key to it and see if the person who sent it really was the person you were expecting, because only two of you would know what that key is.

This means that you're not only able to verify that the data has not been change, but now, you know for sure who sent this data. It is absolutely verified just based on the hash. Again, we're not changing anything with the text or the document or the original piece of information that was sent. You don't need fancy, asymmetric encryption. This is a simple symmetric key. You're using the same key on both sides to be able to determine this information.

This is commonly seen, actually, in **IPsec**. It's commonly seen in **TLS**, which is the big brother now, the new version of **SSL**. And it's a simple process to simply add this key to a very standard set of paddings and implement that within the message to create the hash. It actually is one where you have multiple passes to finally come up with what the final hash might be. So you reverse this process on the other side. You simply go through the same thing. If you end up getting exactly the same hash at the end, then you know the other side had that same secret key, and you can feel very good that the person who sent this is now verified.

**Tags:** [802.11](#), [certification](#), [comptia](#), [cryptography](#), [security](#), [wep](#), [wireless](#), [wpa](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Symmetric Encryption Ciphers – CompTIA Security+ SY0-401: 6.2**

The speed of symmetric encryption makes it a good choice for our high-speed networks. In this video, you'll learn about **RC4**, **DES**, **3DES**, **AES**, **Blowfish**, and **Twofish**.

Let's look at a number of symmetric encryption ciphers. As you recall, symmetric encryption means that we're using exactly the same key to encrypt the data as we are to decrypt the data. The first one we'll look at is **RC4**. This stands for **Rivest Cipher 4**. Some people may say it stands for **Ron's Code 4** because this was written by Ron Rivest, one of the fathers of cryptography, or at least modern cryptography.

This is also, as you recall, part of the WEP standard. This was the wireless encryption that we used that we really don't use any more because of all the problems associated with WEP, and part of those problems were related to vulnerabilities and issues in the **RC4 methodology**. **RC4** has what we call a **biased output**.

And here's the interesting part of this spelled out. If the third byte of the original state is 0 and the second byte is not equal to 2, then the second output byte is always 0. And by having these types of discoveries in the cipher, itself, we were able to find that, perhaps, it was not as secure as we would like it to be.

So although it was, for short time, very useful to be able to use over our wireless networks, once we started digging in and discovering these little problems, we quickly realized this was not going to be the symmetric cipher that we wanted to use for our wireless networks or really for much of anything anymore. So you don't really see RC4 around much anymore. Even our wireless networking, we decided to take and change the algorithms in our wireless encryption to be the WPA2 standard. And in that, we moved completely away from RC4 and began to use the AES Cipher.

Another common set of symmetric key ciphers that you'll see out there is both **DES** and **Triple DES**, and you'll occasionally see **Triple DES** abbreviated as **3DES**. This stands for the **Data Encryption Standard**. It's one that was created between **1972** and **1977**, specifically for the **National Security Agency in the United States** by **IBM**. This was something that they wanted to create as a standard for the entire government, and they did. This became part of what they called the **FIPS standard**, or the **Federal Information Processing Standards**, and it's one that you still see around, being used, perhaps not the **DES part**, but certainly, the **Triple DES part**.

**DES** was a **64-bit block** cipher that used a **56-bit key**, and that's a very important part of this. DES is a very, very small key to be able to use this. And as our processing power has gotten better and better and stronger and stronger and faster in modern times, we have found it very, very simple to be able to crack, to brute force, a DES key, and because of that, we've decided not to use DES any longer. In fact, it's really hard to find a technology still using DES. You could crack a DES with a mobile phone these days. It's painfully easy.

So what we've decided to do instead is use Triple DES. **Triple DES** takes that same idea of DES and really does the same encryption three times, and in each case, you could be using three different keys every time. You could be using one key on the first pass, a different key on the second pass, and then, back to the first key on the third pass. Or maybe, you just use the same key all three times to be able to encrypt this. This makes it harder to do the brute force. It takes a lot longer to be able to try to figure out what the original key might be, and these days, we are really seeing Triple DES in a lot of the products we use. Although, many people have even realized this is getting a little long in the tooth.

We would really like to use **AES** for what we're doing, and that stands for **Advanced Encryption Standard**. It's really one of the most modern symmetric key ciphers out there and one that you'll see in a lot of different places. **AES** became part of the **FIPS standard** in **2001**. The Federal Government decided that the Advanced Encryption Standard was the one that they would like to go with into the future. It took them five years of evaluating different types of ciphers to finally standardize on AES.

And it was created, interestingly enough, by two Belgian cryptographers. Here's their names. I'm not even going to try to pronounce their names. But you can see that the effort that went into getting a particular type of cipher that would be very secure and very flexible for the **Federal Government** was extremely important to them. This is a **128-bit block** symmetric cipher, and they have different key sizes that you can use, anywhere from a **128-bit up** to a **256-bit key** size on both sides of this symmetric cipher.

You'll see this used in WPA2. When we moved from the WEP encryption to WPA2 on our wireless networks, AES was a big part of that encryption standard. Two rather significant symmetric key ciphers are **Blowfish** and **Twofish**. You may have heard these before because they are very open. Anybody can take advantage of them. Blowfish was created in 1993 by Bruce Schneier, a very well-renowned security expert. It's a 64-bit block cipher, and it can have a variable length key, anywhere from 1 bit up to 448 bits. And it's been a very secure set of encryption. Nobody's really been able to break all 16 rounds of this encryption, and it's still even being used today in many applications.

What's also interesting about **Blowfish** and about **Twofish** is there are no patents associated with this encryption algorithm. A lot of the earlier encryption algorithms had patents associated with them. You had to pay a licensing fee to be able to use them, but this was specifically created to be in the public domain, which means anybody can take advantage of this. Twofish came after Blowfish. It is the successor to Blowfish. It uses a **128-bit block** size, and it can have key sizes up to **256 bits**.

And many people contributed to this particular algorithm. The effort was to make it even stronger, even better than Blowfish. And again, they've still got a set of algorithms, a set of ciphers, that nobody's really been able to find any big problems with, and it's still being used today. Again, there's no patent associated with Blowfish or Twofish. These are in the public domain, and so anybody can be able to take these particular algorithms and use them in their development, use them in their applications, without any type of licensing whatsoever.

**Tags:** [3des](#), [aes](#), [blowfish](#), [certification](#), [ciphers](#), [comptia](#), [cryptography](#), [des](#), [encryption](#), [rc4](#), [security](#), [symmetric](#), [twofish](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Asymmetric Cryptography Algorithms – CompTIA Security+ SY0-401: 6.2**

Asymmetric cryptography has enabled secure communication, digital signatures, and much more. In this video, you'll learn about the well-known asymmetric algorithms of RSA and Diffie-Hellman.

One of the internet's most popular encryption algorithms is **RSA**. This **RSA** stands for **Ron Rivest, Adi Shamir, and Leonard Adelman**, and it was published back in 1977. It uses this is public-key cryptography system, where you have a public key and a private key to be able to encrypt information, decrypt that data, and even digitally sign information. This is based on an idea on finding the product of two extremely large, prime numbers. You have to be able to understand those two factors to be able to decode all of this information.

That's a very simplified way of describing how this works, but it does provide us with some great capabilities of sending encrypted data across the network and ensuring that it will be secure from end to end. The RSA algorithm used to be proprietary, but now, it's been released into the public domain. And we use it extensively on our websites to do SSL, or what's now called TLS types of encryption. If you see anybody using a public or private-key encryption mechanism on the internet, there's a good bet that they're using RSA.

Another algorithm that is used to protect keys as they are exchanged over the network is called the Diffie-Hellman key exchange. This is the idea of being able to send keys across the network, but still be ensured that nobody's going to be able to use those keys to decrypt your private information. This key transfer mechanism was published in 1976 by Witfield Diffie, Martin Hellman, and Ralph Merkle, and one important consideration here is this was really designed to transfer keys across the network. It's not, by itself, a method of encrypting or authenticating people's communications over the network. This is something that simply allows us to send keys from one end to the other and still protect that key and the method that's going to be used for encrypting data using those keys.

It's very common to see Diffie-Hellman key exchange used in things like Perfect Forward Secrecy, which is a way to transfer keys and encrypt information on web servers. This uses is **Ephemeral Diffie-Hellman**, which means those keys will only be used for a short period of time. You'll commonly see that written as **EDH or DHE**, and it's combined with the elliptic curve cryptography to be able to do the encryption. So if you ever see a server and it's using a key exchange method of **ECDHE**, it's really referring to this method that's used that we commonly call **Perfect Forward Secrecy**.

**Tags:** asymmetric, certification, comptia, cryptography, Diffie-Hellman, RSA, security

**Category:** CompTIA Security+ SY0-401

## **One-Time Pads – CompTIA Security+ SY0-401: 6.2**

Encrypting with a one-time pad is a very strong encryption technique. In this video, I'll demonstrate how you can use a one-time pad to encrypt your data.

A one-time pad is a cipher that was created in the early 1900s, and it was built when teletype machines were first becoming popular as a way to encrypt the communication on teletype. So this was all done on pieces of paper that would go into a teletype and pieces of paper that would come out on the other side. It was an automated system. It was one that really had a very interesting effect on the communications because then, you could really have private messages go back and forth between one place and another. And it really worked on this concept of the pad, and if you think of the pad as a pad of paper, that's really what this ended up looking like is a single pad of paper with a key imprinted upon it.

This was really interesting in that it wasn't complicated, there wasn't a lot of mathematics involved, and it was one that was also very, very secure. When used properly, a one-time pad is one of these unbreakable kind of ciphers, and as we get into understanding more about the one-time pad, you'll start to understand why it would be so difficult to break this type of communication. For the one-time pad to be this secure, there were a few rules we had to keep in mind. The first one is the key, the piece of information that is on our pad of paper, needs to be the same size as the plain text that we need to encrypt, so the number of letters in the key and the number of letters in the message you're sending are exactly the same. Just keep that in mind.

The second rule is that the key is really completely randomized. This is not a pseudo-random or some type of a very static computer function that's creating this. It really is what we call a true random set of characters that we're putting on there, or a set of numbers. A one-time pad can be used in many different ways. The key should only be used one time, and that's one of the nice things about having this on a piece of paper. We use the key. We encrypt with it. On the other side, we decrypt with it, and then, we throw away the key. And you pull off that piece of paper on the pad, you burn it, you get rid of it, and there's obviously another key you would need to use next time.

That's one of the important parts of this is every time you send a message, the key is going to change, thereby making the entire communication very, very difficult to decrypt. Even if you were able to crack the key one time, you would not be able to crack it again because now, the key is completely different. There are, hopefully, only going to be two copies of this key, one on the person who is sending the message, one the person who is receiving the message, and those are the only two people who would ever have a copy of this key. If somebody was to get a copy of the key somewhere in the middle, they would absolutely be able to decrypt this. So if you follow these rules, you can be assured that your one-time pad communication is not going to be seen by anyone else.

The process of encrypting with a one-time pad is relatively simple. We're going to step through it right here. Obviously, we would follow these same steps in reverse to decrypt the information. The first thing we want to do is put our entire alphabet down, and we're going to assign every letter a number. The easy way is to start at zero with A and end up at 25 with the letter Z. That will be— at least the numbers, we'll be able to use to perform our calculations.

Now, let's take a message. Let's take something in plain text like the word "hello," and we would like to encrypt this. But to encrypt it, we're also going to need a key, and as you recall, we need a key that's exactly the same size as the plain text. So if we go to our one-time pad and we look at our key, we see that our key, in this case, X, M, C, K, L, a random set of letters. Obviously, this key will change every time we send a message. So

we could send the word “hello” this time. The next time we send the word “hello,” it’s going to be completely different in the cipher text that we look at because your key is going to be different every time.

Well, we can’t calculate or perform any type of mathematics on letters, so we need to convert these to numbers. And of course, we have our conversion chart right here at the top. So let’s convert “hello” into a series of numbers, 7, 4, 11, 11, 14. And let’s take the same thing with our key and convert that, 23, 12, 2, 10, 11.

Now, we’ve got two numbers, and we’re just going to add them together, and if we add 7 and 23, well, we kind of go off the end here to 25. If you hit 25, you go all the way back to zero and start counting up again. So 7 plus 23 happens to be the number 4. We’re going to associate this with a letter in a moment.

So if you add all of these columns up, you get 4, 16, 13, 21 and 15, and if you, then, convert those back to letters, you get E, Q, N, V, Z. So there’s our encrypted message. The idea is, on the other end, someone will have the exact same key that we have. They’ll take our message, simply subtract the numbers from it to come up with the plain text numbers, and then associate those back with the letters H, E, L, L, O, to get the message “hello.”

**Tags:** [certification](#), [comptia](#), [cryptography](#), [one-time pad](#), [pad](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **NTLM – CompTIA Security+ SY0-401: 6.2**

NTLM has been used to encrypt user authentication details in the Microsoft operating systems. In this video, you’ll learn about the history of NTLM and how the password information was stored in Windows.

An authentication method that was created by Microsoft is one called **LANMAN**. You may see this referred to as **LAN Manager**. This is one that was created from Microsoft and 3Com. They were just getting into the world of network-based operating systems. So this is well before Windows NT. This is well before any of our Windows Vistas or Windows 7s. This was a very, very early type of operating system, and it’s one that still needed a way, of course, to authenticate.

So Microsoft has their own challenge response system in LANMAN. And it was very similar to what we saw with CHAP, but it was a little bit different. For instance, it was only uppercase ASCII characters, that you can only have a maximum password size of 14 characters, and notice that if you had passwords over seven characters, it split those off at seven. So if you had an eight-character password, what you really had was and seven-character and another one-character password that it would save. So already, the information that’s being stored is not, perhaps, the best one to have there, from an encryption perspective.

And the passwords are not salted either, which means that they’re always going to look exactly the same every time. We’re not adding a bit of randomness into the password process when we’re sending it across the network. So there’s challenges there with keeping that secure. There needed to be different ways to look at handling this LANMAN configuration, so Microsoft tweaked it just a little bit to try to make a few things more secure.

The update to that to make it more secure came with Windows NT, and this was updated to something called NTLM, NT LAN Manager. This is what was used in early versions of Windows NT. The password is now Unicode, which means it has a lot more flexibility on the types of characters you can have in there, could be up to **127 characters long**, and

it's stored as a **128-bit MD4 hash**, which is generally a lot more secure than the smaller DES hashes that were being used in the LANMAN configuration.

But even that wasn't good enough. A new version called NTLM version 2 came out. This came out with Windows NT Service Pack 4, and this added some additional security. We had a new password response. There was an MD4 password hash, the same as what we had with the NTLM version 1, and there was a hash of the username and server name combined with that in there. So now we had a little bit more information thrown in there to make this a little bit more encrypted as it went across the network. This means it's not going to be exactly the same every time. There's going to be some things that are going to be randomized when that information is hashed and sent across the network. And there's also this variable-length challenge sent that has a timestamp, some random data, some domain name information, a little bit more details in there so that we could make that conversation a little bit more secure during the authentication process.

There are few vulnerabilities you should probably be aware of when dealing with NTLM version 1, NTLM version 2, and part of this was created because there were two versions of this NTLM authentication. Therefore, we had to make sure that older systems could authenticate. If an older system didn't know how to authenticate with NTLM version 2, then it may be locked out of the system. So unfortunately, a number of legacy systems kept not just the NTLM version 2 password, which was pretty secure, but it kept the older, insecure NTLM version 1 password as well. The challenge is that if somebody gained access to that NTLM version 1 database, they would be able to have a much easier way to decrypt and figure out what people's passwords were.

We also had a problem within NTLM where it was vulnerable to what we call a credential forwarding attack. What this essentially meant that we could use the credentials of one computer to gain access to another computer. Now obviously, that's not what you want to have happen as well. Microsoft has introduced bug fixes and updates to their operating systems to avoid these types of situations, but it's something you should be aware of if you're using a number of these legacy authentication systems.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [lanman](#), [microsoft](#), [ntlm](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Transport Encryption Algorithms – CompTIA Security+ SY0-401: 6.2**

We use a number of algorithms to maintain the security of our data as it flows across the network. In this video, you'll learn about SSL, SSH, and IPsec encryption protocols.

One of the most common encryption methods on the internet revolves around this trio of algorithms **SSL**, **TLS**, and **HTTPS**. And we often refer to them as exactly the same thing, which is the encryption that we use in our browser. But really these are different types of algorithms that were created at different times. But there is a relationship between them.

SSL stands for secure sockets layer. It was one of the very first encryption mechanisms that we had in our browsers when we were using the internet. It was created by Netscape. So that was built into one of the very, very early browsers that we were using in 1996.

What we found was that there were a number of things that could be improved upon. And we created a new version that was more public, one that was a standard, one that wasn't specific to Netscape. And we called this transport layer security. It was derived from SSL. It is the newer version, if you will, of SSL. But now it's a standard. It's a worldwide standard. And you can find it in RFC 2246.

And when you start combining TLS, this mechanism for security encryption, inside of a browser or inside of the protocols that go back and forth between your browser and the web server that would be HTTPS, or your HTTP protocol that is secure. The s stands for

security. And that uses this TLS, which we often just generically refer to as SSL, to encrypt all of that communication that takes place between your browser and between the web server.

If you looked at the protocol itself it would run over TCP port 443, by default. Obviously it doesn't have to use that. But that's where we most commonly see it. It's built into every browser you'll find these days so that we can have these encrypted methods sending information back and forth. We can put our credit card in a browser. We can send private information back and forth to our health care company over the web. And we can really be assured that nobody can look into that data and see the information that's inside.

**SSH** stands for **secure shell**. And this is a very common encrypted method of communicating to a server through a terminal screen. Being at the command line typing in information you'll very often see this communicating over **TCP port 22**. Now the **SSH**, or **secure shell mechanism**, is one that we can also apply to more than just typing things in at a terminal screen like this. We can use this to be able to do, of course, the remote administration. We can also do file transfers with this encrypted methodology, what we call **SFTP** for **secure file transfer**. And we can do **SCP**, **secure file copy**, to send information back and forth. But when we need to have that access to a computer but we're not really going through a web browser, we're not using HTTP, we're not using that SSL TLS type configuration we can use secure shell to make sure that all of our communication is absolutely encrypted.

The **SSL**, **TLS**, or the **HTTPS** in our web browser is good encryption. The **SSH** that we might use in a terminal screen provides us even other types of encryption. But what if you needed any type of data to be encrypted? What if it wasn't necessarily in a web browser? What if it wasn't necessarily at a terminal screen or doing file transfers? What about all the other things that we do online? What about all the other things that we do on the internet?

Well for situations like that we have another type of encryption mechanism. This is called **IPsec**. This is a type of security that was built specifically for **TCPIP**. And it's designed to work at that Layer 3, to work with IP packets. This IPsec mechanism allows us to have confidentiality and integrity in the communication that we have between devices, or between hosts, on both sides.

We have built in encryption into IPsec. There's the ability to even sign every packet in IPsec. So when we receive a packet we can be assured that is exactly the same packet that was sent from the other side. This is an extremely standardized mechanism of transportation. You will see this on routers, on firewalls, on the clients that move between those devices, communicate between those devices. It's something you'll find in RFC 4301 through RFC 4309. There are many, many standards built into the IPsec mechanism.

There are two core protocols we can look at within IPsec that make all of this work. The first one is called authentication header. You'll see that abbreviated as AH. The other protocol is one called encapsulation security payload. We almost always refer to that as ESP.

One of the challenges in setting up communication between two devices across a network is making sure that both sides can understand each other and both sides are using the same keys. And that's, of course, our challenge is making sure that we're able to communicate what key we're using without actually sending the key information back and forth, especially not over the network or someone would be able to see that. So there are two phases in setting up a communication back and forth between two devices via IPsec.

The first is called **phase one**. And in phase one we do something called **internet security association and key management protocol**, which is abbreviated **ISAKMP**. We often

just abbreviate this as the internet key exchange. You'll see it abbreviated if you're configuring an IPsec client or an IPsec device you'll see it abbreviated as **IKE, or Ike**. This is the process where both sides identify themselves. They recognize each other. There may be some security built in to what we're looking at on both sides of the communication. And they exchange keys back and forth. There's a couple of mechanisms that can be used to do that. This is usually a protocol that's running over UDP and port 500 on UDP. If you're looking at a network decode you'll be able to see it in there with those packets going back and forth.

Once phase one is in place, the keys to being exchanged, now the second phase can begin. In the second phase, which we call quick mode, we're simply setting up and communicating to each device what ciphers we understand, what protocols and key sizes we would like to use to be able to talk back and forth. Both sides will coordinate that process. They'll decide on which ones they would like to use. And then after phase one and phase two are up and running the communication is now built between those IPsec endpoints. And they can communicate securely between them.

As we mentioned earlier, there are two protocols in IPsec. The first one we'll look at is the authentication header. This is a hash that is created that's based on the packet and the shared key that both sides of the IPsec communication are aware of. They usually will use something like **MD5, SHA 1, or SHA 2** as common hashing mechanisms. And what it does is, in the normal packet, where you have an **IP header**, you have a **TCP UDP header**, and then there's data, it'll stick this authentication header right here in the middle and send the packet to the other side.

On the other side the same hash is done of this data and the shared key. And it compares what it comes up with versus the authentication header that was sent. This is that digital signature I mentioned that's created in every packet. So when the other side gets this packet it knows it's received exactly what was sent to it to begin with.

Of course, with IPsec the encryption piece is what a lot of people are looking for. And it's that second protocol of the encapsulation security payload that provides that encryption. This is providing both a hash and an encryption of this data, the hashing done with **MD 5, SHA 1 or SHA 2**. The encryption part is usually done with Triple **DES** or **AES** to make that happen.

You'll see included to the packet a header, a trailer, and what we call an integrity check value to this. So you'll see in the **ESP** packet there will be your normal IP header. There will be your **ESP**, the **encapsulation security payload** header. There will be your TCP UDP information, some data, then the trailer, and the **ICV**. This encapsulates everything together sends it across down the line. And we know at that point not only is the data encrypted but we're able to do integrity checks for it as well to make sure that we receive it in exactly the form that it was sent.

Some uses of IPsec are between a client and a server, where we're simply talking back and forth. And we're taking our existing IP packets. And we're putting the secret IP information inside of our existing packet. And that is a transport mode of IPsec where we're using our existing header, our existing TCP UDP, and data header, and information. And we're simply adding the transport information around that. So we're sending the packets back and forth in the normal way. It's just everything within the packet now gets encrypted and protected using IPsec.

But there are some connections with IPsec that are what we call tunneled mode, where we will have two devices. This is usually done with two endpoints, like two firewalls. And an encrypted tunnel will be built between those firewalls. And all information between point A and point B will be encapsulated inside of this tunnel and sent to the other side. And on the other side it's pulled out of the tunnel and placed back on to the network.

And that is one where there is a new IP header that's built. Everything is jammed into that packet, encrypted with IPsec, and sent across to the other side. So whether you're using a transport mode IPsec or a tunneled mode IPsec the result is that you're going to get information across the network. It's going to be encrypted. You'll be able to check the integrity of that data and be assured that everything you're sending back and forth is protected.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [https](#), [IPsec](#), [security](#), [ssh](#), [ssl](#), [tls](#)

**Category:** [CompTIA Security+ SY0-401](#)

## **Strong vs. Weak Encryption – CompTIA Security+ SY0-401: 6.2**

Not all encryption algorithms are alike, and some are much better than others. In this video, you'll learn how to evaluate encryption strength and how developers can use the `bcrypt` library to generate secure hashes.

Through the years we've used many different kinds of encryption algorithms to be able to protect our data, and as the years have gone on some of these algorithms have become easier and easier to crack, whereas others have maintained their strength. And we usually categorize these as strong cryptography and weak cryptography.

Of course, regardless of the algorithm that's used to encrypt this data, everything is always subject to brute force checking. So if somebody's trying every possible key to try to find a way to get into your data, they can certainly go through this brute force method. But, of course, you can make keys that are so large that it would be functionally impossible to be able to try every possible key in the limited amount of time that we have here on Earth.

These strong algorithms have been here for quite some time. That's the reason we call them strong algorithms. Other algorithms, like the **Wired Equivalent Privacy** or **WEP** that we used to use on our wireless networks, had design flaws associated with it that we found after it had been implemented. That went from being a strong encryption algorithm, and overnight turned into one that was remarkably weak.

Some strong encryption algorithms that you'll find out there are things like **PGP** or **AES**, whereas weak encryption algorithms might be things like **WEP**, which of course had that design flaw, or something like DES where you had very small **56-bit keys**.

These days we use triple DES and larger keys and have effectively turned that into a much stronger algorithm by modifying some of its underlying algorithm features. There is a way that you could use some of these weak keys to create a little bit more of a stronger key. By itself the weak key doesn't protect you very well. It's probably a very short key, and it's subject to a lot of brute force attacks.

But you can make a weak key stronger by performing multiple processes to the same key. For instance, you could take and hash a password, and then you can hash the hash of that password, and then hash the hash of the hash of that password and so on. This is called key stretching, or key strengthening, and the only way you could then brute force that particular key is to reverse each one of those hashes to get back to the original key.

This means that if you wanted to brute force the original key, you would have to reverse that hashing every single time to try to get back to the original key. This certainly limits how much a bad guy would be able to do when he was brute forcing. It would add a lot more work to his plate. And ultimately, you're still using a very small key, you're simply stretching it out to provide you with more security.

If you are an application developer, you might want to use these key strengthening or key stretching algorithms, and there's no reason to rebuild it from scratch. There are many libraries out there that you could use.

One very popular library is called **bcrypt**. This creates those hashes from passwords. You'll find this in the **UNIX crypt library**, and it uses the Blowfish cipher to be able to do those multiple rounds of hashing one after the other after the other.

Another pre-built library you might find is the **Password-Based Key Derivation Function 2** or **PBKDF2**. This is part of RSAs key standards, and you'll find it written up in **PKCS #5**, and it's also part of **RFC 2898**.

If you are an application developer, there's some already pre-built ways that would allow you to keep your information secure so that you can do your development work and not have to worry about all of the underlying cryptographic algorithms.

**Tags:** [bcrypt](#), [certification](#), [comptia](#), [cryptography](#), [hash](#), [security](#), [strong](#), [weak](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Certificate Authorities – CompTIA Security+ SY0-401: 6.3**

Our browser encryption relies on certificate authorities to maintain the trust of your certificates. In this video, you'll learn how certificate authorities are used on our computers and the differences between a **commercial CA** and a **private CA**.

The use of a **certificate authority** is what builds the trust inside of our browser when it begins encrypting data to a third party website. That's the thing that we want to watch for whenever we're sending encrypted information that we're not only protecting the data, but we really are exchanging that encrypted data with the right person on the other end. And it's these certificate authorities that allows us to do this.

The way that this works is that if you need a certificate you go to any one of the certificate authorities. And if you were to look at your browser, you would see behind the scenes listed in all of the certificates is all of these CAs that your browser trusts. So if I go to any one of the CAs and I ask them to provide me with a certificate they will digitally sign it and send it to me. I go through this process of creating a key pair I send my public key to any one of these CAs.

They then confirm that it's really me. They go through a verification process then they digitally sign it and send it back to me. That means that anybody who then hits my web server can see that it has been digitally signed by a certificate authority that is already trusted inside of the browser. There's also a number of different trust levels that the CA can provide. You'll notice when you go to some websites that your browser bar will turn green.

And other websites it turns green and gives you additional verification options. So the CAs can give you, of course, different types of certificates back that provide more, and more, and more security. And they usually will step through a number of different checks to make sure that the person who's receiving the certificate really is the person on the name of the certificate.

If you're an organization that has a lot of internal servers that you would like to be able to encrypt data back and forth, you may want to go to an external certificate authority and have to pay them every time you want a certificate. Instead you may want to build all of your certificates in-house. And you would do that by creating your own certificate authority. You would simply set up a certificate authority server in your organization and you would sign all of your internal private keys.

The other side of this, of course, is that you have to take the certificate authority information and install that or push that out onto all of the desktops who will be accessing those servers so that they will trust the servers just as they trust servers that have been signed from commercial certificate authorities. Generally, you're configuring and setting this all up with things like **Microsoft Certificate Servers** or **OpenCA**. There are a lot of different ways to build your own certificate authorities in-house. And if you have a lot of servers and you need to provide that level of encryption with those certificates then you can save a lot of money doing this in-house rather than going outside to a commercial certificate authority.

**Tags:** [certificate authority](#), [certification](#), [comptia](#), [cryptography](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Key Revocation – CompTIA Security+ SY0-401: 6.3**

An encryption key may need to be revoked for many reasons. In this video, you'll learn about automated key revocation and how to revoke keys in a web-of-trust.

There are times when we create a certificate and then decide maybe we would not like to use that certificate any longer. There is a process for key revocation and this is handled through a **CRL**, which stands for a **Certificate Revocation List**. This is usually maintained by a **Certificate Authority**, so you can go to one central place and view the CRL for that particular CA. There's lots of reasons you might want to revoke a key– maybe the key is not one you'd like to use anymore, maybe you feel that the key has been compromised and you want to be sure that nobody can use that key to do any type of encryption.

A significant case for key revocation was in April of 2014– and if you want to look up the CVE in 2014 of 0160 you can read all about heartbleed. This was the name that was associated with a bug that was discovered in a very popular web encryption library called **OpenSSL**. This **OpenSSL** flaw allowed people from the outside to view the memory of a machine and be able to pull private information directly from that device. Some of that private information could even be the private key for the web server. And because you never knew if somebody did actually gain access to your private key, every web server had to have a replacement certificate.

And now that you've got a new certificate, you wanted to be sure that nobody would try to use the old certificate and so everybody had their old certificates added to the **CRL**. To be able to query the **CRL** and look at the revocation list, a protocol was created called **OSCP**– stands for **Online Certificate Status Protocol**– and your browser knows how to use that protocol to communicate back to the **CEA** and check the status of a certificate.

These messages are usually sent via HTTP so it's very easy to get that across the network and through firewalls. And it's a way that is very standardized across many different browsers. However, many browsers don't have the knowledge of how to use **OSCP**– especially older browsers were never developed with **OSCP** as part of its configuration.

And even today some modern browsers– even though they are **OSCP capable**– have not been configured in the software to even check for revocation. The concept of checking for revoked keys is one that we didn't pay a lot of attention to, but now that we've discovered this heartbleed problem– and now that we have so many keys that had been revoked– a lot of browser manufacturers are going back and doing the development work to properly support this very important **OSCP protocol**.

In a web-of-trust you don't have a Certificate Authority. If you're using something like PGP the keys are managed by all of the key users. You would then go to a friend of yours and have them sign your key and you would sign theirs. And then their friends would sign theirs, and there would be a web-of-trust created all the way through all of those signed keys.

If you then need to revoke a key, you would create a revocation certificate. And you can keep that revocation certificate on some centralized key servers that are located on the internet. You can also configure some of this software to have other people create revocation certificates for your key that way you're all working together to make sure that the keys that are in use are really the ones that are appropriate and trusted for every user.

### **Digital Certificates – CompTIA Security+ SY0-401: 6.3**

We rely on digital certificates for much of the encryption that we use over the Internet. In this video, you'll learn about digital certificates and what information is contained in a digital certificate.

We've talked a lot so far about digital certificates, but what are those really? What is this digital certificate? This certificate that we would use in our browsers, that we would get from a web server— those are public key certificates. It allows us to take a public key that's out there and have it associated with this digital certificate. It's a way to communicate this information in a standard form wherever you happen to be.

This digital signature that we might put on a digital certificate also adds some trust. So you might have in your Public Key Infrastructure a certificate authority that's signs this digital certificate and that adds additional trust to it. If you're in an environment where you're using something like **PGP**, or **OpenPGP**, that is something called a web-of-trust to make sure that everybody trusts everyone else.

This certificate creation is one that is usually built into the operating system, especially in Windows. Then Windows server— you can get something called the **Windows Certificate Services**— and you can automate the process of having these certificates created. This domain services allows us to automate this and stored in a very, very simple way. But if you're using another operating system like **Linux or Mac OS X**, you'll find there are other third party certificate management softwares out there. You can find many of them open source so that you can have digital certificates whatever your operating system might be.

Digital certificates are constructed in a very standard way so that you can use a digital certificate that was created on one machine and other machines would be able to understand that very standard format. That format is called **X.509**. You'll see people referring to their **X.509 certificate**. And there are different versions that you can see for the different types of **X.509 certificates**, but they're generally all following something like this standard format. There may be serial numbers, signature algorithms who issue the digital certificate, some validity frames, a subject, a public key, and also extensions. And all of this information is put in a standard format in that **X.509 certificate** so that you can share it with anyone.

That extension piece that you can have at the end of your **X.509 certificate** allows you to include a lot of different capabilities for that cert. And you'll find that there's an extension ID, whether it's true or false, in this critical field and the value, the string value of what that extension happens to be. So you can have a digital certificate and you can add some extensions on it to say that this is going to be used to digitally sign documents— maybe it's used just for key exchange, maybe this is used by the certificate authority for certificate signing.

So having that extension piece on there really allows you to build out certificates with very specific functions and be able to label it that way. And you'll find when you start building certificates in your operating systems with your certificate authorities, you'll have a lot of these options available to you. It really extends the capability of what you can do with digital certificates.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [digital certificate](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Public Key Infrastructure – CompTIA Security+ SY0-401: 6.3**

If you have a large number of devices using a public key infrastructure, you'll need a way to manage all of the keys in the infrastructure. In this video, you'll learn about public key infrastructures and the key management lifecycle

This entire section of the Security Plus exam is on Public Key Infrastructure, but what is a **PKI**? What is this really consist of? A **Public Key Infrastructure** is not just one single thing. It's a mixture of a lot of things all working together– it's policies, it's procedures, it's hardware and software, and people– all put together to create a standard way to distribute these certificates– to manage them, to create them, to store them, revoke them.

If you're getting into doing anything related to public-key cryptography and you're creating a Public Key Infrastructure, then you're going to be creating something that's pretty big. And it's pretty important and you want to plan it out from the very beginning, and set all of these processes in place so they can be as successful as possible. This **PKI** is going to be responsible for building these certificates and then binding them to people, or binding them to resources. This is the Certificate Authority that's doing this and there is an entire section of what we'll talk about based on the trusts that are created between that Certificate Authority and the people that are using these certificates.

In a Public Key Infrastructure there is an entire life cycle that revolves around the keys. Obviously where we start with is creating the key to begin with. We are creating a key with a particular cipher, with a particular key strength, or key size. It's one that is very specific and we'll have to make decisions at the very beginning when we first create this key of exactly all of those technical details associated with the key generation process.

Then we'll create a certificate. We will allocate that key to a person. We'll bind those together and create that X.509 certificate that includes the key and all of those other things that we mentioned in our previous video. Then we distribute those keys to the end user and those certificates out to our certificate servers. We need to make sure that process is available to our users and that it is as secure as possible.

This key management lifestyle also includes then storing this information. We're creating a lot of different certificates. We're building out a lot of different keys. Some of these are extremely valuable, extremely private keys. We want to make sure they cannot be used for unauthorized use and so there is a very important storage mechanism we have to have in place for that. Ultimately there may be a need to revoke these keys.

Keys might be compromised– part of the business might shut down, people may leave the organization– or they just maybe a certain amount of time to that key is valid. And so at the end of that time frame, or for one of those other reasons, we need to have a process in place that is able to properly revoke those keys and make everyone realize and understand that we have revoked them. Perhaps have some key revocation lists or other mechanisms in place so that people understand which keys have been revoked and which ones have not. And finally, an expiration.

Keys may only have a certain shelf life. You may have created them to only be valid for 3 months, or 6 months, or 1 year. And at that time the key is no longer valid and you will have to create new keys. You'll see that happen all the time. And that's one of the things about this management life style is then it goes all the way back to the top again where we create new keys because those have expired and we are able to perform the entire

process over again. As long as you're thinking about your Public Key Infrastructure, and building out your processes and your procedures to take into account every single step along the way of this life cycle you should have a very, very successful PKI.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [lifecycle](#), [public-key](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Key Recovery – CompTIA Security+ SY0-401: 6.3**

If you lose your encryption key, you'll lose access to all of your data. In this video, you'll learn about key recovery techniques and how some certificate authorities can manage your encryption keys.

When we're encrypting information we sometimes forget just how important that private key is. If we were to lose that key, we would no longer have access to any of that data. So that key becomes extremely valuable to us. And the idea of key recovery means that we've already put some processes in place to make sure that should something happen to that key, we still have a way to recover that data that might be encrypted.

One of the ways to do it is to back it up, but of course one of the challenges you have is you don't want to make too many backups of your private key. You don't want too many versions of that private key getting into other people's hands, so therefore you want to be sure that it's backed up or, perhaps, not backed up too much, but don't wait. You still need to have at least 1 backup of that key. So don't think that not backing it up is even more secure. You need a happy medium there so that the key is protected— maybe it's stored away, could be in a safe somewhere— but it's not in a place where someone may be able to run across it or at least get access to that private key.

In many organizations there is already a key recovery process that was thought from the very beginning. May everybody realize that if you lose a hard drive or you lose that private key, we still want the organization to have access to that data. So the entire process of recovering that is probably built into your PKI. It may be something that's just done automatically every time a set of keys is created.

This is something that would need to be planned for though. You need to already have it in place prior to rolling out these keys so that you can then recover them. There are a couple of different approaches to doing this. One is to take every key you create and just back them up. That way if you lose them somewhere you can always go back and recover them with the backups that you have maybe stored in a safe somewhere.

Another way to do this is through the process of this public key encryption— have an "M of N" control, which means that you would have to have at least a certain number of people all contributing together to be able to recover certain amount of information. So that adds a little more confidentiality to it. And it ensures that no 1 person might be able to get this secret key and get into other people's information without everyone else the organization know that that was going on.

If you're already using a certificate authority, some of this recovery process is already built into it. So although it sounds very difficult to implement, it may just be if you check marks on a screen to make that happen. This may be a little more difficult to do with private certificate authorities because very often you're building a certificate, or having them build you a set of certificates. They're sending you the private key and the public key.

And they tell you right away, if you lose this, it's gone forever. We're not keeping a copy of it. You can't recover it from us. There's no way to do that. So we're handing it to you and it's now completely up to you. You best put that in a very safe place because if you

lose it, you're going to have to come back and create brand new keys all over again. This entire key recovery process is an extremely important one. And if you're building out a certificate authority, you're creating entire PKI for your organization, it's certainly going to be something you want to look at.

**Tags:** certification, comptia, cryptography, key recovery, security

**Category:** CompTIA Security+ SY0-401

### **Public and Private Keys – CompTIA Security+ SY0-401: 6.3**

For asymmetric encryption, you'll need the public and private key pair for successful encryption and decryption. In this video, I'll demonstrate the creation of a public and private key pair.

The implementation of your public and private key creation is something that's usually part of a formal process, especially if you have a formal certificate authority set up. It's integrated into the security policy. You know exactly how to request. You know the process that goes on to get registered and have that key and the certificate provided back to you. It may be something that is very, very structured. And you need a lot of documentation, and you need to show up in person.

And it has to be linked to your ID that you would use, or it might be more relaxed. It might be something like PGP, or open PGP where you are outside of an organization, and maybe you're building out a certificate just for your own use. Let's build out a **PGP** secret key and public key so that you can see what the process is to go through. There is a front end to the **open PGP** standard, called **GPG**.

It stands for **GNU** privacy guard. And you can download **GPG** for a **Mac OS 10** that I'm running here. You can run it on Windows, you could run it on Linux. There are command line options to run there as well. So you can really have that capability on many different operating systems. And what I'd like to do on my GPG is build out a brand new public and private key pair so you can see the process I went through when I first created the key pair that I have here.

So I'm going to go up to the menu that you can't see on the screen to the key pulldown menu, and choose to generate a new key. And the options I have available are to create a key type that is **RSA** and **RSA**. That looks good to me. Let's specify a particular link. Let's call this one a **4,096 byte key**, a big long one. This key will not expire. And I'm going to put in a different email address in here that's something that is a test at professors-messier.com.

And I could even put in my comment here that this is a test key, do not use. I plan on deleting this when we're done. But if this key happened to be posted to the public key server, people would see that comment and realize, this was not a key that would be normally in use. I choose generate key, and one of the messages that comes up is that there needs to be a lot of randomization here.

That's because the random number generators that are in computers are really only pseudo random. There's a way to predict certain things that those random number generators might do. So it's saying, move your mouse around. Have some keyboard access, do things that would create something that's a little more random than what we would find in the chips. We also need to enter a passphrase. And that's because this private key that we're going to create is going to be a very, very important key.

If we have this private key, we can obviously decrypt everything that's sent to us. So as an extra layer of security, PGP asks for us to be able to enter a passphrase here. So I'm going to put in a passphrase. This one's not very long. But I'm going to put it in any way. And it says in fact, it says my passphrase should be at least eight characters long. And it tells you, and this should be part of the CA process that you would set up, it tells you that this is not secure at all.

Do you really want to take this one anyway? Yes, I would. And it's going to ask me again to input that exact same passphrase so that we know we've got the right one there. At this point, we're going to continue to generate this information. It's building out the keys for us. And eventually, it's going to create that and put it into our key ring. After moving my mouse around and creating a lot of randomization, finally we got all the information we needed to create this brand new key that has my test account associated with it.

And you could see, it's created a **4096 byte key**. That is an **RSA key**. It has a short ID, a very specific ID number. And there's the fingerprint of the key. Now, behind the scenes of course, this private key is on my hard drive. And if I wanted to, I could export that key and have a look at it right on the screen. Let's do that. In my GPG program, I have the option to export that particular key pair. And I can specify to not only export the public key, which is something that I might want to provide to everyone, but I can also export the private key so you could see how that looks on the screen.

Now, you'll see the public key is this bunch of text. This is the key itself. I could even copy and paste this entire begin **PGP public key block** all the way to the end, and put it online for people to download. And they could drag it, and drop it into their key ring as well. I'm going to move down just a little bit in here so you could see. I'm going to keep going. These are big, **4096-bit keys** as we go through. And at the bottom of this list, you're going to see, there's also the end of the public key block.

And I also specified for this to output, the private key block. Now, obviously you would not do this normally unless you were planning to take that private key and put it somewhere safe. You don't want to share this with anyone. They might be able to figure out your passphrase, and then be able to use this. But notice that the private key, very similar to the public key in the way that it's structured.

It's again, just a lot of text that you could drag and drop into one of these key rings. That means that this information is very easy to move between systems. It's just a bunch of ASCII text. But both the private and the public key, of course, are mathematically linked. So we know that we can take anything that is sent to us that has been encrypted with that public key, and decrypt it with the private key that we've created at the same time.

**Tags:** certification, comptia, cryptography, private key, public-key, security

**Category:** CompTIA Security+ SY0-401

### **Key Registration – CompTIA Security+ SY0-401: 6.3**

If a key is going to be associated with a person, there must be a formal way to validate the association. In this video, you'll learn about the best practices for registering and assigning encryption keys

The role in your Public Key Infrastructure that ensures that you have the right people associated with the right certificates is called the Registration Authority. It's this registration process that ensures that you have exactly the right people lined up with exactly the right certificate. You don't want someone receiving a message from James signed by James and you find out later it was somebody who had James' certificate.

It's this registration process that is in place to ensure that you don't have that type of fraud or any type of mix up with those certificates later on. This can be done very casually. You can call somebody on the phone. You can have somebody login with certain credentials. Or there may be many, many steps that someone has to go through to finally be registered properly in your Public Key Infrastructure.

To give you an example of one type a process in place for key registration, let's look at the Federal Public Key Infrastructure Policy Authority. This is the X.509 certificate policy for the US Federal PKI Common Policy Framework. And there is the URL. And we have

very, very detailed processes and procedures here in the United States. And this is a document that describes the processes and procedures around having a key associated with a person.

And if you start reading through this, there is some very, very interesting level of details about what's involved. If you go to Section 3.2.3.1, The Authentication Of Human Subscribers, then you can see what is required. You may have to have your identity verified no more than 30 days before a certificate is issued. You may need agency approval— you will need agency approval. It may require in-person appearance. You have to have a verification of employment.

You have to have your government ID with you. It may also require biometric data. It may require verification of your credentials. You may need to bring in a credit card or utility bill of some kind. There is 1 or many of these that might be required based on the level of security associated with you and the certificate that you're trying to get. So you can see just how important that key registration process is and how detailed you can really be to make sure that that certificate is matched up properly with that user.

**Tags:** certification, comptia, cryptography, key registration, security

**Category:** CompTIA Security+ SY0-401

### **Key Escrow – CompTIA Security+ SY0-401: 6.3**

The escrow of encryption keys can be a necessary process, but it isn't without controversy. In this video, you'll learn the advantages and disadvantages of escrowing your encryption keys.

With key escrow we're taking our decryption keys and putting them in the hands of a third party. That means all of our private keys will be held by someone else. And that means that we're planning in a case where we might need to decrypt some of that information that you might have. This could be an absolutely legitimate process that you have in place to ensure that you always have access to your data.

For instance, a business may need to access information that an employee had encrypted on their hard drive of their laptop, and that employee may no longer be with the organization. So you need some way to get back in to that laptop. If you're a government agency, you may need to decrypt data that might be coming from partners. And so you might have a third party hold the decryption keys so that there's somebody who is an independent agency that would control and make sure that was not abused.

And it could be conceived as controversial. If you're storing your private information and it's your data that's being encrypted and stored on a laptop or on a computer, you may not feel very happy about third parties having access to that. But in the United States, at least, organizations who have distributed laptops to their employees own all of the data on those laptops. So it may be just a normal part of doing business and certainly an absolutely legal part of doing business, so that you can have access to the data on that laptop regardless of what might happen to the private keys or to the employee that happen to be using that laptop.

The process of having a key decrypt information is relatively straightforward. With key escrow a lot of the work is done on the process itself. You want to be sure that it's very clear that you have already in place— a set of procedures so that there's no questions about what the process is if you ever need to take advantage of that key escrow. These keys are very, very valuable. They're very important and you should absolutely have this in place before you start that escrow process.

You also have to be able to trust the people you're giving these keys to. You want to be sure that if those keys are stored in a certain place that they're going to be protected. You

want to be sure that nobody can have access to those keys who should not have that access. So obviously this is not something you do on a whim. It's not something you do without a lot of process and a lot of procedures in place.

And just to make sure that all of these particular conditions are controlled. If you need access to certain data on someone's laptop, there should be a series of documentation and communication in place so that you're able to show that. In some cases it may take legal proceedings to get the data and provide access to that encrypted information. So it's going to be very important that you plan ahead so that all of those different contingencies are taken into account.

**Tags:** [certification](#), [comptia](#), [cryptography](#), [key escrow](#), [security](#)

**Category:** [CompTIA Security+ SY0-401](#)

### **Trust Models – CompTIA Security+ SY0-401: 6.3**

There are many ways to manage and validate the trust of our encryption keys. In this video, you'll learn about using **CAs**, **mesh relationships**, **web-of-trust**, and **mutual authentication**.

One of the most important aspects of your **Public Key Infrastructure** is that of trust. You have to be able to be assured that the certificates that you're using are those that you can trust, and that the names associated with that are the names associated with people who might be receiving those and might be decrypting the information that you're sending to them. Depending on the type of infrastructure you have and the way that you've built out your Public Key Infrastructure, there may be a number of different models that you use for trust. For instance, if you have a single certificate authority you might find that everyone is receiving all of their certificates from this one place, and you can trust now that that one certificate authority is managing that process for everyone.

If it's a large organization though, you may find that it's easier to have more than one certificate authority and you might be spreading the trust out a little bit. And then that environment is a hierarchical trust relationship where you might have a single root certificate authority. It might be issuing certificates to the intermediate certificate authorities, and then even underneath those you could have leaf certificate authorities, and finally your users and your resources.

So it depends on the size of your organization, and you may need that. There may be geographical requirements. There may be structural requirements within your organization that would require that level of control and the ability to spread the trust around from the very top root all the way down to the other certificate authorities. There's also a type of trust called the mesh trust relationship, where every certificate authority trusts all of the other certificate authorities.

And that works extremely well if you have 2, or 3, or 4 CAs. But as you can see, once you start adding more, and more, and more, and you add one more in every single one of those certificate authorities, now all has to trust everyone else— when you start adding them— becomes a little bit more difficult. And at a certain point you simply can't scale any larger. It's just too complicated and too difficult to manage. If you've worked at all with PGP, or with OpenPGP, then you know the trust relationship there works a little bit different.

You don't have a centralized certificate authority, what you have is everybody trusting everybody else. So it's a friend of a friend of a friend, and everyone would sign everyone else's certificates. Very often you have a certificate signing party— have all your friends in a room— everybody signs everybody certificates. And eventually those types of things now go out to other areas and you eventually can see people you've never met before,

but you happen to know in between someone who is signed a certificate that's common to both of you. And therefore there's additional levels of trust that you have associated with that.

This trust relationship in **OpenPGP**, and **PGP**, is one that what we really build ourselves. It's a person-to-person trust and it's one that works a little bit better when you have something that's completely uncontrolled and uncentralized like the PGP web-of-trust infrastructure. Another type of trust relationship would be a Mutual Authentication, where the server authenticates to the client and the client authenticates to the server, and both of those entities trust each other exactly the same amount.

Whether you're using the single certificate authority, the web-of-trust, a mesh type of trust relationship— doesn't matter— as long as you're using the one that works for your environment. And that's the key. As long as the trust is there, you can be assured that the certificates you are using are ones that you can rely on everyday.

**Tags:** certification, comptia, cryptography, hierarchical, mesh, security, single ca, trust models, web-of-trust

**Category:** CompTIA Security+ SY0-401