



Practical Network Defense

Master's degree in Cybersecurity 2020-21

Penetration testing and vulnerability assessment

Angelo Spognardi
[*spognardi@di.uniroma1.it*](mailto:spognardi@di.uniroma1.it)

*Dipartimento di Informatica
Sapienza Università di Roma*



Principles of penetration testing

Vulnerability assessment vs. Penetration testing

- Vulnerability assessment
 - the process of identifying and quantifying vulnerabilities within a system
- Penetration testing
 - the process of evaluating the security of a computer system or network by simulating an attack from malicious outsiders and malicious insiders
 - active analysis
 - active exploitation
 - intent is to identify and demonstrate consequences of security weaknesses
 - should not disrupt the network...

Penetration testing steps

- Planning
- Non-intrusive target search (intelligence gathering)
- Intrusive target search
 - Scan!
- Data analysis
- Threat modeling and Exploitation
- Reporting
- Repeat! → maybe in six months...

Planning

- Scoping and terms
- IP addresses and domains
- Rules of engagement
 - DoS?
 - Social engineering?
- Times, targets, etc.
- Identify goals

Intelligence gathering

- Gather as much information about your target from publicly available information as possible
- Non intrusive search → reconnaissance
 - Use public resources
- Many tools available
 - nslookup, whois, traceroute/tcptraceroute, fierce.pl
 - google hacking
 - business research
 - online resources:
 - www.serversniff.net
 - The Way back machine (www.archive.org)
 - Shodan (www.shodan.io)
 - Recon-ng

Intrusive target search

- Probe and explore the target network
 - Remember: explicit written permission! Otherwise, it is ILLEGAL
- Aims:
 - Find live systems
 - Discover open ports
 - Discover services
 - Enumeration
- All this can be done by network scanners

Threat modeling and vulnerability scan

- It can start once we have the enumerated services
- Identify assets (what are we trying to get?)
 - Data
 - Employees (executives, administrators, etc.)
- Examine gathered information and look for exploitable known vulnerabilities
 - This can be automatized with tools like:
 - nessus
 - nexpose
 - OpenVAS (open source)
 - Nikto (open source)
 - Burp suite
 - Acutenix...

Vulnerability assessment

- Typically conducted remotely
- Usually conducted from outside the network
- It could also be performed within the network
 - Inside NVA
- It could also be performed within single hosts
 - Local NVA
- Typically remote assessment is done first

Data analysis

- Understanding the obtained data, acquired with the previous steps
- Determine missing elements or impartial results
- This step can also be repeated after the exploitation
- Can be time consuming

Exploitation

- The process of validating a discovered vulnerability
 - Can also be unsuccessful because the vulnerability can not have exploits
- Again, we can use a tool for executing the exploit:
 - metasploit
 - BeEF
 - Throwback

Reporting

- Document everything that can describe the steps that lead to the exploit
- It will be the reference to address found issues
- Include:
 - Description of the exploited/exposed vulnerabilities
 - Analysis of the finding extracted from the reports
 - Recommendations of patches, best practices and solutions to mitigate or remove the risks
 - References to further readings
- Example:
<https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf>

Methodologies

- Different processes and methodologies are available to perform security assessments
 - OSSTMM
 - NIST SP 800-115
 - for a (non-exhaustive) list, check [OWASP website](#)
- Provide a standardized approach to conduct penetration tests
- But... attackers don't agree on the methodologies and, more important, don't follow the rules!!\

Open Source Security Testing Methodology Manual

- An OSSTMM audit is an **accurate measurement of security** at an operational level that is void of assumptions and anecdotal evidence.
- **Consistent and repeatable.**
- As an open source project, it allows for any security tester to contribute ideas for performing more accurate, actionable, and efficient security tests.
 - Version 4 still draft

OSSTMM 3

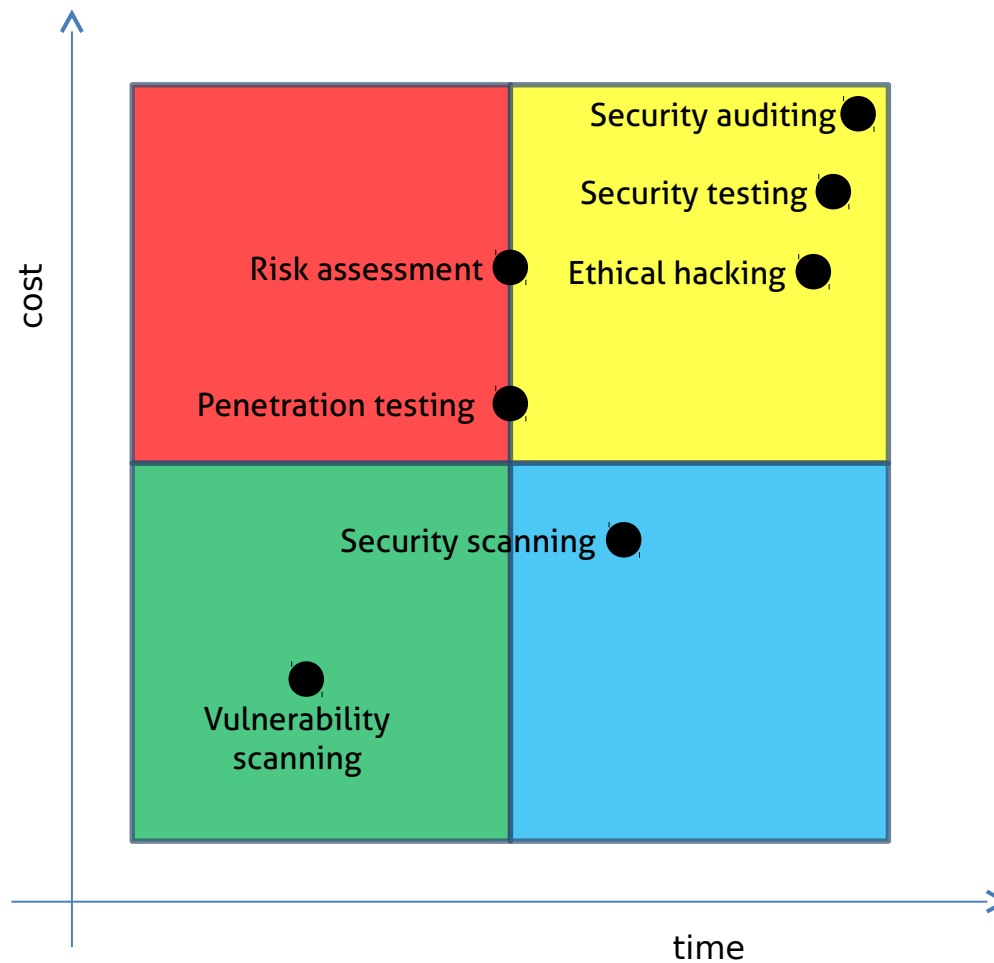
The Open Source Security Testing Methodology Manual
Contemporary Security Testing and Analysis



Created by Pete Herzog
Developed by ISECOM

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

Types of security test



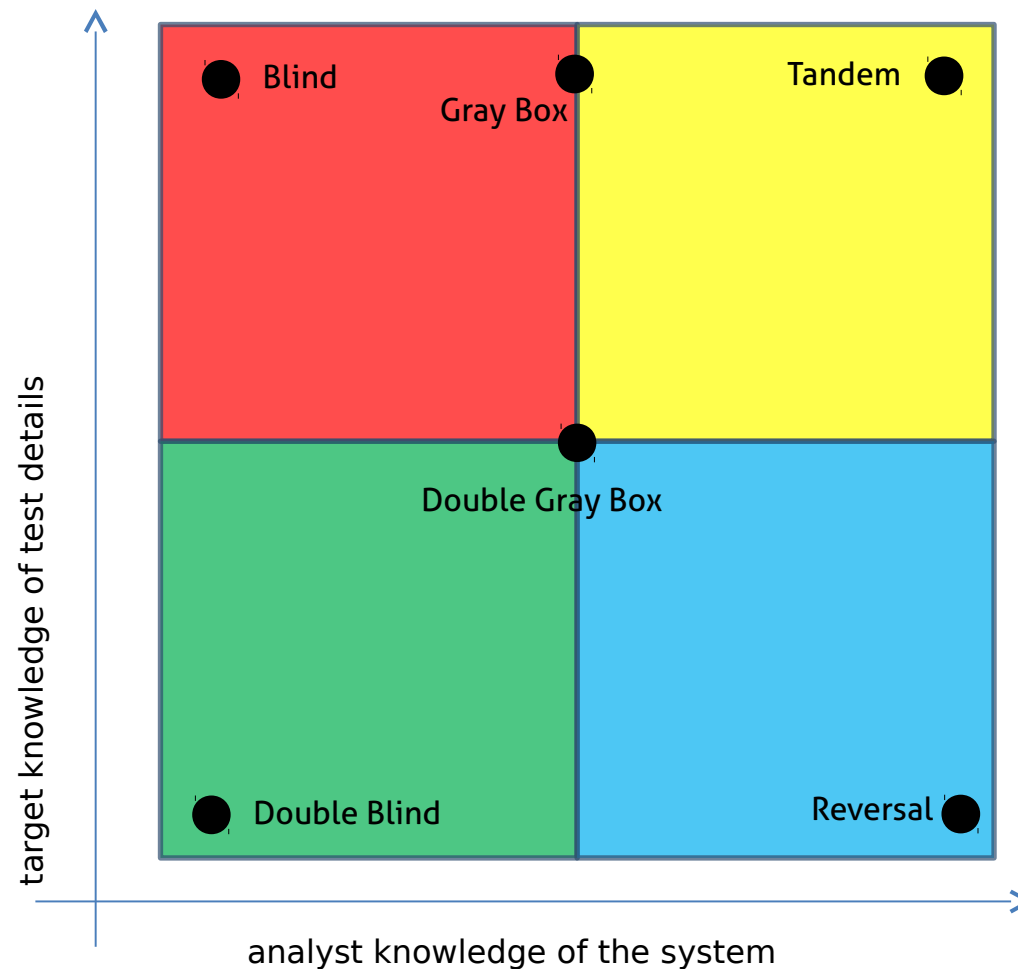
Security tests

- **Vulnerability Scanning:** **automated** checks for known vulnerabilities against a system or systems in a network.
- **Security Scanning:** vulnerability scans which include **manual** false positive verification, network weakness identification, and customized, professional analysis.
- **Penetration Testing:** a goal-oriented project which simulates an attack from a malicious hacker. It includes gaining privileged access by pre-conditional means.
- **Risk Assessment:** security analysis through interview and mid-level research which includes business justification, legal justifications, and industry specific justifications.

Security tests 2

- **Security Auditing:** a hands-on, privileged security inspection of the OS and Applications of a system or systems within a network or networks.
- **Ethical Hacking:** a penetration test of which the goal is to discover security flaws in the system and the network within the predetermined project time limit.
- **Security Testing:** and its military equivalent, the Posture Assessment, is a project-oriented risk assessment of systems and networks through the application of professional analysis on a security scan where penetration is often used to confirm false positives and false negatives as project time allows.

Security test typologies



Typologies

- **Blind:**

- The analyst engages the target with no prior knowledge of its defenses, assets, or channels.
- The target is prepared for the audit, knowing in advance all the details of the audit.
- A blind audit primarily tests the skills of the analyst. The breadth and depth of a blind audit can only be as vast as the analyst's applicable knowledge and efficiency allow.

- **Double Blind:**

- The analyst engages the target with no prior knowledge of its defenses, assets, or channels.
- The target is not notified in advance of the scope of the audit, the channels tested, or the test vectors.
- A double blind audit tests the skills of the analyst and the preparedness of the target to unknown variables of agitation. The breadth and depth of any blind audit can only be as vast as the analyst's applicable knowledge and efficiency allows. This is also known as a **Black Box**.

Typologies 2

- **Gray Box**

- The analyst engages the target with limited knowledge of its defenses and assets and full knowledge of channels.
- The target is prepared for the audit, knowing in advance all the details of the audit.
- A gray box audit tests the skills of the analyst. The nature of the test is efficiency. The breadth and depth depends upon the quality of the information provided to the analyst before the test as well as the analyst's applicable knowledge.
- This type of test is often referred to as a **Vulnerability Test** and is most often initiated by the target as a self-assessment.

- **Double Gray Box:**

- The analyst engages the target with limited knowledge of its defenses and assets and full knowledge of channels.
- The target is notified in advance of the scope and time frame of the audit but **not the channels tested or the test vectors**.
- A double gray box audit tests the skills of the analyst and the target's preparedness to unknown variables of agitation. The breadth and depth depends upon the quality of the information provided to the analyst and the target before the test as well as the analyst's applicable knowledge. This is also known as a **White Box** test.

Typologies 3

- **Tandem**

- The analyst and the target are prepared for the audit, both knowing in advance all the details of the audit.
- A tandem audit tests the protection and controls of the target. However, it cannot test the preparedness of the target to unknown variables of agitation.
- The true nature of the test is thoroughness as the analyst does have full view of all tests and their responses. The breadth and depth depends upon the quality of the information provided to the analyst before the test (transparency) as well as the analyst's applicable knowledge. This is often known as an **In-House Audit** or a **Crystal Box** test and the analyst is often part of the security process.

- **Reversal**

- The analyst engages the target with full knowledge of its processes and operational security, but the target knows nothing of what, how, or when the analyst will be testing.
- The true nature of this test is to audit the preparedness of the target to unknown variables and vectors of agitation. The breadth and depth depends upon the quality of the information provided to the analyst and the analyst's applicable knowledge and creativity. This is also often called a **Red Team exercise**.



Vulnerability assessment

Penetration testing steps

- Planning
- Non-intrusive target search (intelligence gathering)
- Intrusive target search
 - Scan!
- Data analysis
- Threat modeling and Exploitation
- Reporting

What is a Vulnerability?

A weakness which can be exploited by a threat actor, such as an attacker, to perform unauthorized actions within a computer system

[http://en.wikipedia.org/wiki/Vulnerability_\(computing\)](http://en.wikipedia.org/wiki/Vulnerability_(computing))

What is a Vulnerability Scanner?

Something that Discovers and enumerates vulnerabilities on systems or applications.

-Q – Does a Scanner Protect my network?
-A – No.

http://en.wikipedia.org/wiki/Vulnerability_scanner

Vulnerability scanners strength

- Very good at checking for hundreds (or thousands) of potential problems quickly
 - Automated
 - Regularly
- Can help identify rogue machines
- Helpful in inventory devices on the network
- Provide a generic risk level
- Explain why the item is a risk
- Provide detailed information on how to apply remediation
 - The differences of how your scanner does the above items are some of the key differences between the scanners

Types of vulnerability scanners

- Agent vs. agent-less
 - Agents are monitoring software running on each end node and communicate with a central scanner
 - Highly customizable and very precise information
- Passive vs. active
 - Passive scanners rely on monitoring network traffic and are completely unobtrusive
 - Can complement Active scanners
 - Real-time monitor of new entities in the network or configuration changes

Challenges of Vulnerability scanners

- Typically slower than simpler port scanners
- Some scanning / testing may disrupt operations (DDoS testing)
- False positive rate may be high and require human judgment
 - However false positives are preferred over false negatives
 - Better alert on something that's not there than miss stuff
- Vulnerability DB must be updated frequently
 - Always playing catch-up to changing threats
- Security resources are often decentralized
- The security organization often doesn't own the network or system
- Determining if the fix was actually made
 - Ignoring it → accepting it

How do vulnerability scanners work?

- 1) Discovery
 - Check to see if the target is responding to network traffic
- 2) Port Scans
 - Find what ports are listening on the system
- 3) Service Detection
 - Based on the ports listening, talk to them to “guess” what they are based on patterns and other information
- 4) Operating System Detection
 - Based on the available information, “guess” the operating system

How do vulnerability scanners work?

CVE-ID	
CVE-2009-3095	Learn more at National Vulnerability Database (NVD) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description	
The mod_proxy_ftp module in the Apache HTTP Server allows remote attackers to bypass intended access re: related to the embedding of these commands in the Authorization HTTP header, as demonstrated by a certain	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list	
<ul style="list-style-type: none">• BUGTRAQ:20091124 rPSA-2009-0155-1 httpd mod_ssl• URL:http://www.securityfocus.com/archive/1/archive/1/508075/100/0/threaded• MISC:http://intevydis.com/vd-list.shtml	

5) Vulnerability Assessment

- Based on the available information, determine the vulnerabilities that may exist on the system
- Often listed by the “CVE” identifier

What is a CVE?

- **Common Vulnerabilities and Exposures**
- A dictionary of common names (i.e., CVEs) for publicly known information security vulnerabilities
- CVE is the industry standard for vulnerability and exposure identifiers
- CVE Entries — also called "CVEs," "CVE IDs," and "CVE numbers" by the community — provide reference points for data exchange so that cybersecurity products and services can speak with each other

<http://cve.mitre.org/about/index.html>



CVE and CVSS

- When a new cyber-vulnerability is discovered, a CommonVulnerability and Exposure (CVE) number is publicly assigned to it by a vulnerability number assigning authority like the MITRE Corporation
- The vulnerability is then analyzed by the US National Institute of Standards and Technology (NIST) whose Common Vulnerability Scoring System (CVSS) provides both a severity score and 8 commonly adopted security attributes of the vulnerability

CVE and CSS of Meltdown

CVE-2017-5754 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

Systems with microprocessors utilizing speculative execution and indirect branch prediction may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis of the data cache.

Source: MITRE

[+View Analysis Description](#)

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



NIST: NVD

Base Score: **5.6 MEDIUM**

Vector: CVSS:3.0/AV:L/AC:H/PR:L/UI:N/S:C/C:H/I:N/A:N

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to help you find more information that would be of interest to you. No inferences should be drawn on account of the fact that NIST has provided this information. There may be other web sites that are more appropriate for your purpose. NIST does not endorse any product or service, nor does it concur with the facts presented on these sites. Further, NIST does not endorse any product or service. Please address comments about this page to nvd@nist.gov.

Hyperlink

<http://lists.opensuse.org/opensuse-security-announce/2018-01/msg00006.html>

<http://lists.opensuse.org/opensuse-security-announce/2018-01/msg00007.html>

CVSS v3.0 Severity and Metrics:

Base Score: 5.6 MEDIUM

Vector: AV:L/AC:H/PR:L/UI:N/S:C/C:H/I:N/A:N

Impact Score: 4.0

Exploitability Score: 1.1

Attack Vector (AV): Local

Attack Complexity (AC): High

Privileges Required (PR): Low

User Interaction (UI): None

Scope (S): Changed

Confidentiality (C): High

Integrity (I): None

Availability (A): None

CVE and CVSS of Heartbleed

🚩 CVE-2014-0160 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

Source: MITRE

[+View Analysis Description](#)

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



NIST: NVD

Base Score: **7.5 HIGH**

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Evaluator Impact

CVSS V2 scoring evaluates the impact of the vulnerability on the host where the vulnerability is exploited. To help you understand the impact of this vulnerability to your organization, take into account the nature of the data that your organization's risk acceptance. While CVE-2014-0160 does not allow unrestricted access, the exploit does leak information from memory locations which have the potential to contain sensitive data, such as cryptographic keys and passwords. Theft of this information could enable other attacks. The impact of this vulnerability would depend on the sensitivity of the data and functions of that system.

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST's webpage. We have provided these links to other websites because they may have

CVSS v3.1 Severity and Metrics:

Base Score: 7.5 HIGH

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Impact Score: 3.6

Exploitability Score: 3.9

Attack Vector (AV): Network

Attack Complexity (AC): Low

Privileges Required (PR): None

User Interaction (UI): None

Scope (S): Unchanged

Confidentiality (C): High

Integrity (I): None

Availability (A): None

Practical example: Buffer overflow

- Aleph One: "Smashing the Stack for Fun and Profit"
- Since then long lists of *exploits* based on buffer overflow
 - 10849 CVE entries match "buffer overflow" (over 135490)!!
- The idea

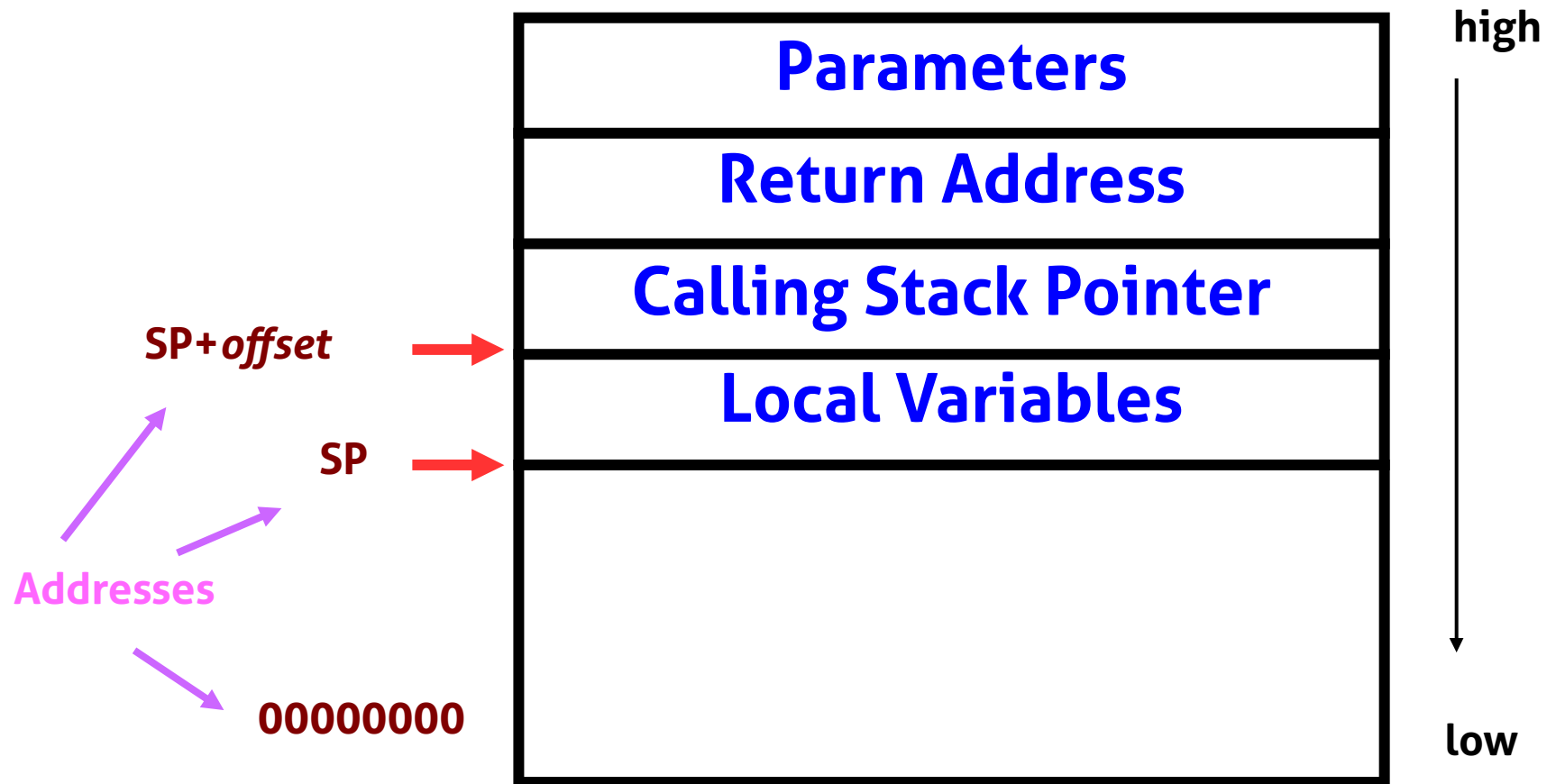
```
void foo(char *s) {  
    char buf[10];  
    strcpy(buf,s);  
    printf("buf is %s\n",s);  
}  
  
...  
  
foo("thisstringistoolongforfoo");
```

Exploit buffer overflow vulnerability

- The general idea is to give servers very large strings that will overflow a buffer
- For a server with sloppy code – it's easy to crash the server by overflowing a buffer (SEGV typically)
- It's sometimes possible to actually make the server do whatever you want (instead of crashing)

The stack frame

For each function call





Stack example

18
addressof(y=3) *return address*
saved stack pointer
y
x
buf

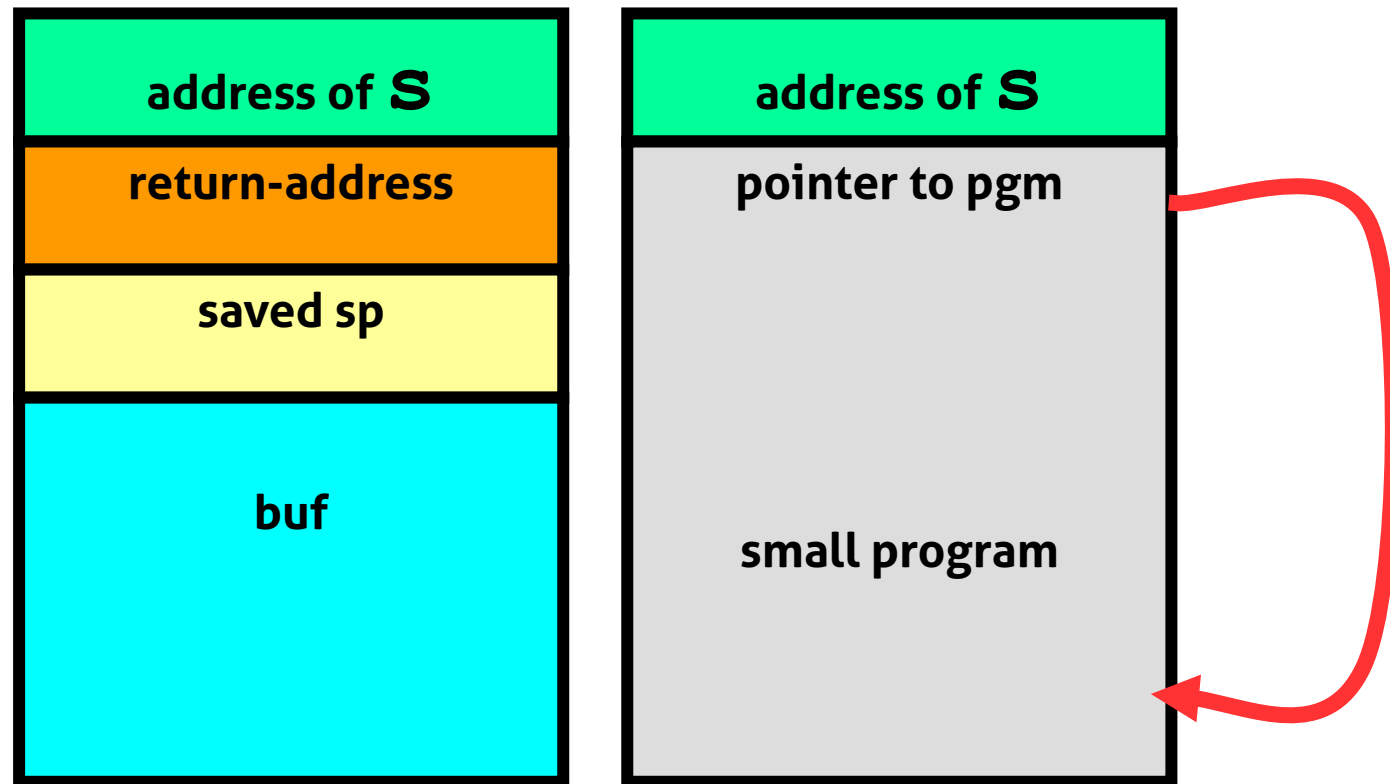
```
x=2;  
foo(18);  
y=3;
```

```
void foo(int j) {  
    int x,y;  
    char buf[100];  
    x=j;  
    ...  
}
```

If the buffer contains a small program...

- Before and after

```
void foo(char *s) {  
    char buf[100];  
    strcpy(buf, s);  
    ...  
}
```



Does it seem easy?

- Many complications...
 - The injection in the stack has to be binary code (not text)
 - Each architecture would need a different binary code
 - You have to guess the pointer where to make the return
 - Many others...
- Some tricks:
 - Shellcodes (programs to strings)
 - NOPs

Many other types of vulnerabilities

- Generally one reason: improper input validation
- But also:
 - Memory exposure/leakage
 - Bad configurations
 - Bad implementations
 - Default credentials
 - Force use of weak/legacy protocols

Vulnerability scanner principles

- Probing specific services/protocols for weaknesses
 - Not just generic IP addresses like “nmap”
- Most useful when working from pre-gathered info
 - From databases or other network scanning
- Similar to virus scanning software
 - Contain a database of vulnerability signatures that the tool searches for on a target system
 - Cannot find vulnerabilities not in the database
 - New vulnerabilities are discovered often
 - Vulnerability database must be updated regularly

Vulnerability scanner methodology

- Manual Attempts and Permutation (fuzzers)
 - Very long time
- Manual Version Probe
 - Slightly better, but still very slow
- Custom Protocol-Specific Attacks
 - requires special knowledge
- Automated Vuln-Scanner
 - simple, fairly reliable, fast, thorough



So You have Vulnerability Data, Now What?

- Vulnerability Data is just a big spreadsheet style list or a report.
- By itself, vulnerability scan data lacks context
 - Can't expect folks to act on 1,000 page reports
- Need to provide some prioritization
 - What are the biggest risks in your environment?
 - What is the level of risk that is acceptable in your environment?
 - What is the threat level that exists in your industry?
- Turn the data into **actionable information**

You start with something like this..



CVE	CVSS	Host	Protocol	Port	Name	Synopsis	Description	Solution
CVE-2003-0831	9	192.168.150.131	tcp	21	ProFTPD File Transfer Newline Character Overflow	Arbitrary code may be run on the remote server.	<p>The remote host is running a version of ProFTPD which seems to be vulnerable to a buffer overflow when a user downloads a malformed ASCII file.</p> <p>An attacker with upload privileges on this host may abuse this flaw to gain a root shell on this host.</p> <p>*** The author of ProFTPD did not increase the version number</p> <p>*** of his product when fixing this issue, so it might be false</p>	Upgrade to ProFTPD 1.2.9 when available or to 1.2.8p
CVE-1999-0502	10	192.168.150.131	tcp	22	Default Password (toor) for 'root' Account	An account on the remote host uses a known password.	The account 'root' on the remote host has the password 'toor'. An attacker may leverage this issue to gain total control of the affected system.	Change the password for this account or disable it.
CVE-1999-0103		192.168.150.131	udp	7	Echo Service Detection	An echo service is running on the remote host.	The remote host is running the 'echo' service. This service echoes any data which is sent to it. This service is unused these days, so it is strongly advised that you disable it, as it may be used by attackers to set up denial of services attacks against this host.	<p>- Under Unix systems, comment out the 'echo' line in /etc/inetd.conf and restart the inetd process</p> <p>- Under Windows systems, set the following registry key to 0 :</p> <p>HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpEcho HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpEcho Then launch cmd.exe and type :</p> <p>net stop simptcp net start simptcp To restart the service.</p>
CVE-1999-0635		192.168.150.131	udp	7	Echo Service Detection	An echo service is running on the remote host.	The remote host is running the 'echo' service. This service echoes any data which is sent to it. This service is unused these days, so it is strongly advised that you disable it, as it may be used by attackers to set up denial of services attacks against this host.	<p>- Under Unix systems, comment out the 'echo' line in /etc/inetd.conf and restart the inetd process</p> <p>- Under Windows systems, set the following registry key to 0 :</p> <p>HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpEcho HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpEcho Then launch cmd.exe and type :</p> <p>net stop simptcp net start simptcp To restart the service.</p>
							When contacted, chargen responds with some random characters (something like all the characters in the alphabet in a row). When contacted via UDP, it will respond with a single UDP packet. When contacted via TCP, it will continue spewing characters until the client closes the connection.	- Under Unix systems, comment out the 'chargen' line in /etc/inetd.conf and restart the inetd process

Using Vulnerability Scan Data

- Analyze the Data and Add Context
 - How are composed the systems and networks?
 - Which are the “controls” in place?
 - The Default CVSS Risk Scoring is scored for systems on the Internet!
- Prioritize the Findings
 - CVSS Scores are a baseline severity
 - Risk & Risk Appetite should vary by company
 - What is most important for the target company?

Actionable information

- Make a “Remediation”, or “Go Fix it” plan
 - Resources are limited & factor into what gets fixed
 - Usually it is impossible to say “Go Fix Everything”, mainly for very big environment
- Can be useful to ask why are the vulnerabilities there in the first place
 - Address the root causes can truly reduce the long term security exposure
 - If not properly managed, the same kind of vulnerabilities will be back!

Trends in Vulnerability Scanners

- More features
 - Configuration management
 - Application & System vulnerabilities
 - Helping Analyze the Data
 - Helping Report on the Data
 - Confirming the Data
- Specialized Scanners
 - ERP Systems
 - SCADA Scanners
 - Database Scanners
 - Application Scanners



Common System Vulnerability Scanners

Nessus
McAfee Vulnerability Manager
QualysGuard
Retina
Nmap
Nexpose
OpenVAS
Core Impact
Saint



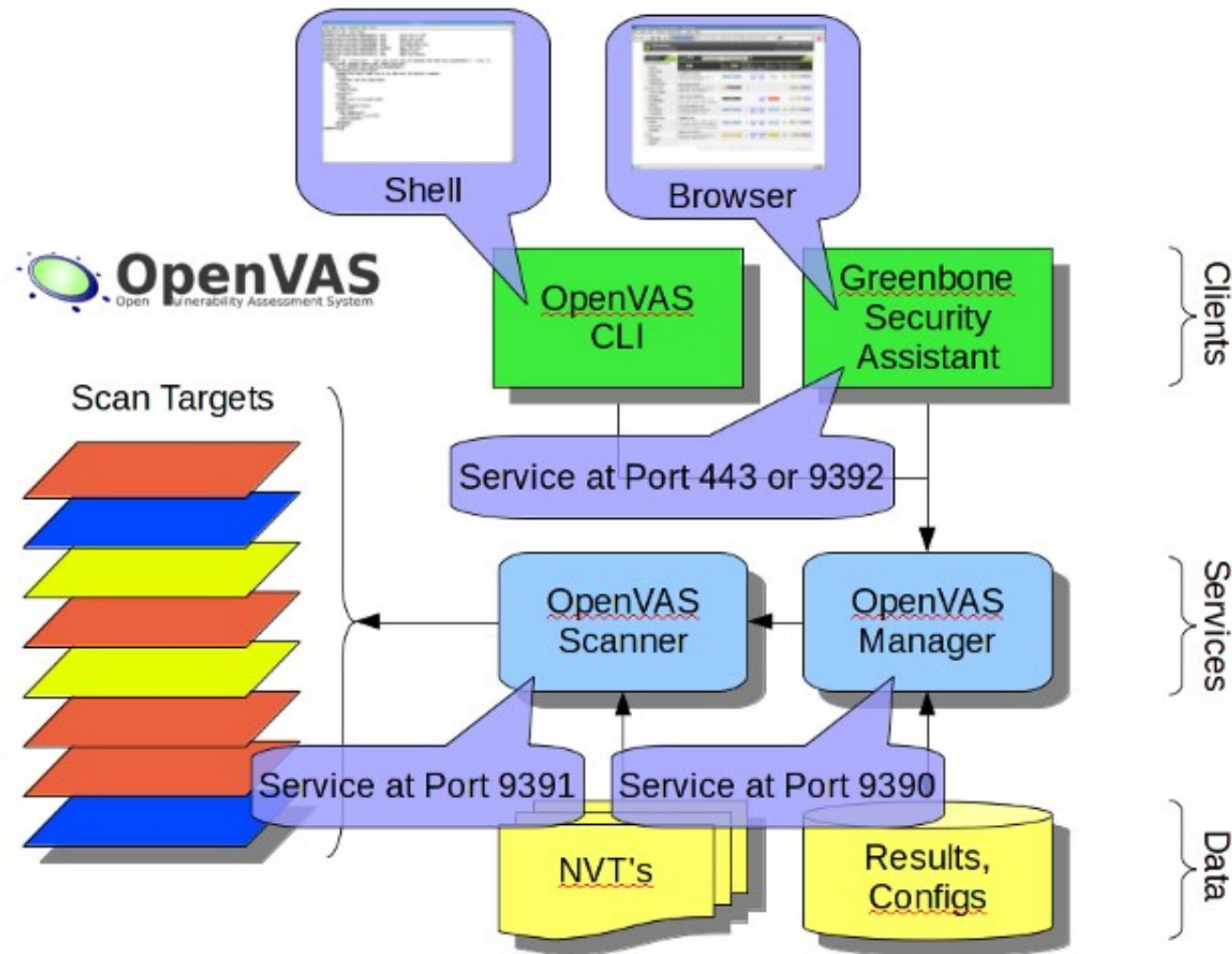
Application Scanner Tools

IBM AppScan
WebInspect
Netsparker
Burp Suite
ParosPro
Nessus
QualysGuard WAS
Zed Attack Proxy
sqlmap
W3AF
Acunetix Freed Edition
Safe3WVS
arachni
Skipfish
N-Stalker
Watobo
SyHunt Mini

OpenVAS

- The open source fork of nessus v2
- Similarly to nessus:
 - OpenVAS can enumerate multiple OSs
 - Identify a computer on the network by using port scanning and zone transfers
 - Identify the OS by conducting port scanning
 - Identify via enumeration any logon accounts
 - Learn names of shared folders by using enumeration
 - Identify services running

OpenVAS architecture



OpenVAS components

- Scanner
 - Scans and receive signature updates of NVTs, Network Vulnerability Tests
- Manager
 - Manages scanning rules
 - Consolidates reports (XML, HTML, LaTeX etc.)
 - Stores settings and history based sqlite
- Clients
 - Command Line and Web GUI (Greenbone Security Assistant)

OpenVAS Client

Greenbone Security Manager

Scan Configs 9 of 9

Name ▲	Type	Family		NVTs		Actions
		Total	Trend	Total	Trend	
Base (Basic configuration template with a minimum set of NVTs required for a scan.)	OpenVAS	2	→	3	→	
Discovery (Network Discovery scan configuration.)	OpenVAS	16	→	2901	↗	
empty (Empty and static configuration template.)	OpenVAS	0	→	0	→	
Full and fast (Most NVT's; optimized by using previously collected information.)	OpenVAS	64	↗	58582	↗	
Full and fast ultimate (Most NVT's including those that can stop services/hosts; optimized by using previously collected information.)	OpenVAS	64	↗	58582	↗	
Full and very deep (Most NVT's; don't trust previously collected information; slow.)	OpenVAS	64	↗	58582	↗	
Full and very deep ultimate (Most NVT's including those that can stop services/hosts; don't trust previously collected information; slow.)	OpenVAS	64	↗	58582	↗	
Host Discovery (Network Host Discovery scan configuration.)	OpenVAS	2	→	2	→	

Greenbone Security Manager (GSM) Copyright (C) 2009-2019 by Greenbone Networks GmbH, www.greenbone.net

That's all for today

- **Questions?**
- See you on MONDAY (Thursday's lecture canceled)!!
- References:
 - OSSTMM
 - <http://cve.mitre.org/about/index.html>
 - <http://www.openvas.org/>
 - Live demo <https://www.greenbone.net/en/live-demo/>