The Unstoppable Force: An Analysis on Distributed Denial Of Service (DDOS) & Address Resolution Protocol (ARP) Poison Attacks

Marco Carvallo, Sean English, Tyler Smith
Florida State University
Team 1

Abstract

Today we use the internet for almost everything. We rely on the services we find online in order to complete the tasks in our daily lives. Because of this information security has really been growing rapidly in the last decade or so. New defence methods are created every day to manage online threats, but there is one threat that to this day causes major problems for even some of the strongest systems in the world. ARP poison and DDOS attack are attacks simple enough for almost anyone to perform, but it is extremely difficult to defend against them. In this paper we will be analyzing these seemingly unstoppable forces and research techniques on how to hinder these attacks. We will be looking at how they function in a safe laboratory environment, how they function and operate as threats, as well as how you can prevent them and defend a service from these kinds of attack. Finally, tactics to protect yourself from other attacks and hardening your network will be discussed.

Keywords: *DDOS, ARP Poison, Information Security, Apache, pfSense, Security Onion*

Introduction

The internet is an ingrained and important part of our daily lives that requires more protection than most realize. The internet can be said to run on a series of networks of varying sizes, yet all of these are constantly under pressure from malicious attacks and threats. One such attack is known as a DDOS attack. The DDOS attack is a dangerous malicious attack that can cripple any network. The ARP poison attack is another one that can further cripple a network's operating efficiency. Our focus will be answering the questions "What are DDOS and ARP poison attacks and how do we stop them?". It is imperative that we understand it in order to prevent it. The purpose of this research paper is to explore the uses of a DDOS attacks and ARP poison attacks in a virtual environment against other networks, understanding their inner workings, and finally how to defend against the attacks. Utilizing this as a basis of our knowledge, we will be able to better explain how to prevent them.

We will achieve this by first going over the initial lab systems settings and setup to give you an understanding of the environment we're working in. This will entail talking about each feature and service inside of the virtual network and how we are utilizing it. These include a kali linux, pfSense firewall, Comodo firewall, Security Onion, Apache web servers, and HoneyBOT/Windows 7 machines, This will be followed by a graphical representation of its topology. Then we will talk about our experiences in dealing with threats from other teams. This entails how we reconfigured our defenses to meet their attacks and a specialized counter attack. Third party programs and packages were also a toolset we utilized to protect our network. One was a vulnerability scanner called Nessus, an antivirus called Malwarebytes, and finally a package for the pfSense firewall called pfBlockerNG. The configuration will be detailed and we will explain how we utilized them in our defense plan.

Following this section, we will then go over the attacks themselves. We will do this by first giving you a basic definition of what a DDOS is. This includes going into the type of tools a DDOS utilizes, some of the legal consequences, the threat types of the attacks, how easy it is to use, how effective they are, and why it is so hard to detect/stop. Then we will go ARP poison attacks, which includes what an ARP table is, how the attack exploits its

functionality, and why ARP poison attacks are so malicious. We will then connect this to our virtual environment and describe how we plan on demonstrating how this attack works, so that we may better understand it. Finally, we will cover how a network can defend against the attacks. This includes suggestions for defending a large scale network (or a small scale network). Defending against this attack also entails suggestions for mitigation and prevention. In DDOS attacks this involves how to use hardware (like firewalls) to help prevent them and what patterns to look for. We also talk about why it is more difficult than normal to stop the DDOS attack. While with an ARP poison attack, it falls into a similar vein of using hardware solutions. Such as firewalls/routing that takes advantage of dynamic inspection. There is also the possibility of utilizing new protocols to help solve the issue of ARP poison attacks. With these ideas and concepts in mind, we hope to fulfill the purpose of this research paper and support our initial statement about the dangers of DDOS and ARP poison attacks. Now, onto some of the pieces of literature and their respective topics that are used in this paper.

Literature Review

Network Protection Mechanisms

An important part of any working business is ensuring that your network has the best protection. While our virtual lab environment didn't have the choice of choosing specific types of servers and firewalls, we were able to do research on the concept to see possible faults our network may encounter. The Security Onion was one of the focal points of our research. The paper "Logging and Monitoring to Detect Network Intrusions and Compliance Violations in the Environment" was a great article that discussed the different roles that Security Onion fills. Security Onion is a intrusion detection system that is also a network security monitor. It contains programs such as Sguil, Snorby, Squert, Snort, argus, and many more. The article goes into detail about how all of these features come into play with a network and provides a topology diagram to show what it is like in a basic environment. Finally, the article goes into detail on how to set up the features and shows how they work with pictures and walkthroughs (Gupta, 2012).

Firewalls are just as important as monitoring software in a network. While we weren't able to choose the firewall, there were online materials that could help us compare the pfSense firewall to other popular ones on the market. The Firewall journal article went over a variety of firewalls and discussed their features. This allowed us to compare the information of the pfSense firewall with those that are commonly used in the industry to help identify possible vulnerabilities or points of concern. It should be noted that this article is from 1997, so its information wasn't taken as up-to-date(Sobol, 1997).

DDOS prevention, detection, and response

This entry focuses on classifying DDOS attacks, preventing them, and how to detect them. According to the author there are five different classifications for the type of DDOS attacks: network device level, OS level, application level, data flood, and protocol feature attack. Network device level attacks aim to exhaust the hardware using exploits such as a password buffer overflow. OS level attacks involve slowing down the actual system which can be done by sending ping attacks which affect the OS. Application level attacks aim to slow down network applications or involve the attacker using applications on the host to slow the system down. Data flooding attacks are the most common and the aim is to take up as much bandwidth to prevent the network from sending and receiving packets. Protocol feature attacks aim to use an exploit in the protocol, an example is using a spoof attack, which is when the attacker alters their IP to make it appear as if a completely different device, or the host IP, is attacking the host.

According to Douligeris & Mitrokotsa, the purpose of the DDOS attack is to temporarily disable the system instead of taking it out completely. Because internet resources are limited, if the attacker has more bandwidth than the victim, the victim is definitely going to be taken out. DDOS attacks have four elements: the attacker, handler of attackers, zombie hosts, and the target. The attacker can also be a handler which uses the zombie hosts to attack the target (Douligeris & Mitrokotsa, 2004). The steps taken in a DDOS attack are as followed: 1. Selection of zombie

hosts, 2. Zombie hosts get compromised, 3. Attacker communicates with the handler to notify them of zombie hosts acquired, 4. the attack commences.

When it comes to DDOS defense there are four categories: intrusion prevention, intrusion detection, intrusion tolerance and mitigation, and intrusion response. There are three locations that defenses can be deployed: victim network, intermediate network, and the source network. In order to prevent DDOS attacks you need to prepare your network by setting up several compliances such as : using globally coordinated filters, disabling unused services, applying security patches, changing ip addresses, disabling ip broadcasts, load balancing, and the use of a honeypot. Anomaly detection and misuse detection are used for intrusion prevention to check if the user is a genuine user. There's not much that can be done to respond to an attack besides looking at logs and using IP trackback to find the source of the attack. According to Douligeris & Mitrokotsa, "research on intrusion tolerance accepts that it is impossible to prevent or stop DDOS completely and focuses on minimizing the attack impact on maximizing the quality of its services"

At the victim network level there is little that can be done since most of it's resources are being attacked. The only thing the victim network can do is recognize that it is getting attacked. The intermediate network level's purpose is to trace back and identify the attacker. The source network is responsible for stopping incoming traffic during the attack but you are limiting all traffic which compromises availability. There is little that can be done to stop a DDOS attack other than limit your own network (Douligeris & Mitrokotsa, 2004).

A Review of DDOS Attacks and its Countermeasures in TCP Based Networks

This section covers various methods of executing a DDOS attack and how they different types of attacks function. The attacks that are covered include: SYN flood attacks, TCP Rest Attacks, ICMP attacks, UDP storm attacks, DNS request attacks, CGI request attack, Mail bomb attacks, ARP storm attacks, Algorithmic complexity attacks, and Spam attacks. Some of these attacks like SYN flood attack, and UDP storm function by overburdening a system's resources.while others like the TCP reset attack function by interfering with the transmission of data to and from the service.

From there the article goes on to explain tools that can be used to issue these attacks. Namely they used Good Bye v3 with a TOR plugin. using this combination of tools attackers are not only able to launch a DDOS attack but they can also spoof the ip of the attacking machine. Next the article explains possible defence mechanisms that can be implemented. To simplify, the author go over the importance of being able to identify an attack and the use of a bloom filter, which is a fast yet flawed filter to check values. Prevention is you best defence since stopping an active attack is very difficult. Through careful monitoring and auditing attacks can be avoided and attackers can even be traced back. Finding the attacker and stopping the attack at the source holds the highest chance of success. If that is not possible a bloom filter will help protect your system by managing transmission in a more organized manner.

Finally the author goes into a Individual Component Analysis, which is a more detailed method of analyzing traffic patterns.

ARP poisoning attacks and their use in script injection

In this entry the main focus is on how ARP based exploits work and how they can be used to insert malicious code into a network or system. According to this journal ARP attacks are carried out by in essence lying to the system. An ARP table will send out ARP request packets to gather information from systems on the network and properly assign them pathways. The table keeps logs of the system's IP address and MAC addresses. The exploit happens when an attacker uses this request to falsely assign themselves to a powerful position in a network. (most often it is the position occupied by the router.)

Once in a seat of power, this article goes on to explain that an attacker can cause a lot of problems for a network. The problem the author focuses on though is the injection of hostel JavaScript code. Because the attacker is now

functioning as the effective router, it is able to send packets back to systems on the network. By injecting hostel code into these packets attackers are able to very discretely infect victim's machines.

In the final paragraphs of this entry the author discusses how to notice the subtle sign that give away this kind of attack. Unfortunately detection requires either very intricate knowledge of the network your using or expensive software. Because ARP based attacks change the IP/MAC pairing it is possible to spot the attack if you can tell that a pairing has been changed. Though because companies can how thousands of pairings, it is unlikely, if not impossible, that a human could remember all of them. Because of this ARP exploit identification software has been created that identifies changes in IP/MAC pairings. The downside is that his software often costs a lot and is rarely implemented.

Detection of ARP Spoofing Attack Using ICMP Protocol

The two techniques for detecting ARP are: Passive technique and Active technique. Passive technique involves constructing a MAC and IP binding table. An alert is raised for a spoofing attack whenever the addresses don't match with the table, you could also set it up to drop all spoofed connections. The problem with this technique is it might be too late before you realize you're getting attacked. The active technique involves injecting a packet in the network that actively interacts with the network.

The ARP spoofing detection system contains the following modules: an ARP packet sniffer module, an IP-MAC mapping database, ARP spoofing detection module, and a response module. Using this model, the packet passes through the sniffer, where the IP-MAC mapping database compares the first entry to the current. The ARP spoofing detection modules receives any entries that are flagged as spoofed by sending ICMP packets to the IP address and compares the response to the initial reply. If this test fails, the response module will alert the administrator of the attack.

This model can be set up using only two rules. The first rule:" The NIC of a host only accepts packets with its own hardware address, broadcast address, and subscribed multicast addresses. The network layer only accepts IP packets addressed to its IP address will drop the other packets silently.". The second rule: "Hosts with enabled IP packet routing will forward the packet to its destination host. All legitimate hosts in the network do not enable IP packet routing and will respond back after it receives an ICMP echo request packet.".

A New Approach to Prevent ARP Spoofing

This journal, "A New Approach to Prevent ARP Spoofing", starts out defining IP spoofing attacking which involve DoS attacks and DDoS attacks. Attacks that involve ARP spoofing include mapping the environment through ARP protocol, ARP spoofing, and ARP cache poisoning which is a man in the middle attack.

MAC spoofing attacks are detected by sending a request of inverse ARP for a mac address or port security but this only works with MAC spoofing.A middleware approach has been proposed for ARP cache poisoning attacks but it doesn't work if the host being spoofed is down or being DoSed. You can always use static IP and MAC binding entries for every host but this network scheme isn't suited for a large organization.A solution that involves public and private key distribution has been proposed for signing ARP messages has been proposed and is referred to as Secure-ARP protocol.

Using the command "ARP -sip_addressmac_address" you can make static entries which prevent attackers from using ARP spoofing techniques on the network. When the addresses are dynamic, it allows the system to accept changes which opens the door for spoofing. This technique stops attackers at the cost of making your network very hard to use for guests and new users.A test is conducted using the statically added mac addresses and the attack is using an ARP poison program called Netcut.This defense was successful against Netcut and it dropped all of the spoofed packets sent.

DDOS and ARP Poison Attacks

DDOS attacks are one of the few attacks that can't be stopped. Even today companies such as Microsoft have had problems in the past with DDOS attacks. A DDOS attack uses zombie (which are usually infected with Trojans) hosts to attack a single target to disrupt a service. A group of these "zombies" are called a botnet. The most common DDOS attacks are bandwidth attacks and application attacks. Bandwidth attacks are used to disrupt network bandwidth by flooding the system with packets (TCP, UDP, and ICMP). Application attacks involve using HTTP errors to overload the system.

Threat type

To really understand a DDOS attack we must first cover what a DOS (Denial of service) attack is. A DOS attack is defined on Wikipedia as "an attack that attempts to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet." While this covers the base concept of this attack it really does not explain it fully. DOS attacks are used to stop the operating function of a webpage, server, or other computing resource by interfering with the transmission of information to or from the service. A very simple metaphor to help understand most of these attacks is to imagine a horse that can only carry 100 pounds down a road. It takes this horse one hour to do this. If you were to come along and load that horse with 200 pounds, it would walk much slower. With all the extra weight it would take the horse two hours now to reach the end of the road. If we were to then put a huge amount of weight on the horse, say 1000 pounds, the horse would be too encumbered and would not be able to walk at all no matter how much time was allotted. A DOS attack follows essentially the same logic. By sending a huge amount of requests or transmissions we can encumber a service to the point that it does not function, or functions so slowly that it is of no use to its users.

The issue with a DOS attack is that if you for some reason cannot encumber the target, the attack will fail. Sticking with our horse metaphor, if someone has a really strong horse that can carry 2000 pounds, and you only have 1000 pounds to weigh it down with, the horse will still be able to move around just fine. Stronger entities like large companies (Google, Microsoft, etc.) will have stronger systems thusly making these attacks ineffective. One person on one computer will never be able to successfully launch a DOS attack on Google.com. This is where DDOS comes into play. While a single person on a single computer can't do much, a single person controlling 10, 20, 100, even 1000 machines can do a lot of damage. Referring back to our super strong horse, even though it can carry 2000 pounds and one person can only load 1000 pounds onto it at a time, with a group of less than five people you could entirely encumber it. Attackers use this logic to effectively take down larger, more powerful targets. According to Akash Mittal (2011) "The current attacks on trendy web sites like Amazon, Yahoo, e-Bay and Microsoft and their resultant disruption of services have uncovered the weakness of the Internet to Distributed Denial of Service (DDoS) attacks."By using malicious software, such as Trojans, attackers are able to gain control over various computers depending on how they choose to infect their victims. Once a machine is infected an attacker uses this system (usually against the owners will) to help attack the target. Each of these infected machines is known as a Zombie and the collection of these infected machines is known as a botnet. With a large enough botnet it is theoretically possible to take down any target because you simply need to reach the threshold of what that system can handle.

Aside from increasing the computing power, controlling this many computers can help an attacker in other ways as well. One of the most common ways to stop a DDOS attack is to simply block all requests and transmissions coming from the source that is attacking you. This can be identified by a few different means but most common way is by looking at the attackers IP address. With only one source this is a very effective way to weaken the attack, but with say twenty or more machines this becomes much more difficult. Large scale attacks have been known to take over hundreds, even thousands of computers making systematically blocking each one a long and difficult process. This also makes tracking the attacker much more difficult. To keep themselves hidden, attackers have been known to attack with their botnet but not with their own machine. This is roughly equivalent to not leaving your figure prints at a crime scene. By not directly involving the main machine, it makes the attack untraceable unless they traced back through one of their zombies. This is also a bit of a double edged sword. Legally someone cannot be arrested or charged with any offence in the United States unless their attack affects more than 10 machines or causes more than

$5000 dollars in damages. So while a larger botnet may make an attacker more powerful, the price for failure is much higher the larger the attack..

DDOS also offers a special feature that other attacks often don't have. It is commonly very hard to tell on more active or popular web pages if you are being attacked, or if you are simply receiving a large amount of traffic. Because most DDOS attacks simulate a service being overloaded by its users, it is easy to mistake high network traffic for an attack and vice-versa. This poses an issue because, as we mentioned before, one of the main ways to stop an attack like this is to block the source from transmitting requests to the service. The danger is that you do not want to block real, authentic, users from accessing the service. In turn, this disguise causes it to be extremely tricky to stop this attack without accidently affecting the legitimate users negatively.

There are two attacks to worry about when defending your system. A volumetric attack and an application level attack. A volumetric attack is very hard to defend against as you're dealing with a botnet but an application level attack uses some sort of open port to take down a service. The most common is a volumetric attack because a host isn't always running services that can be exploited.Volumetric attacks might also use a form of a reflective attack that would spoof their IP, but there are safeguards that can be put up, such as request auditing programs. to decrease the likelihood of a successful DDOS attack.

With our Kali Linux's wide range of scanning and exploit tools, it is the perfect platform for what we have in mind. More specifically we are going to be using the tools nmap and Ettercap. Nmap is a network mapping tool that we will use to scan the network for our opposing teams in our upcoming class exercises.. In the scan we will be given information like other systems IP addresses, what ports they have open and other important networking information that we will need in order to successfully crash their system. Once we have found a main point of weakness we will begin our attack with Ettercap. Ettercap is a network security tool that is capable of doing a number of nasty attacks, the most noted is a MITM (Man in the middle) attack. This attack is where an attacker inserts themselves into the information stream of a service, effectively acting as the router as far as the computer sees. The computer sends information to the attacker now instead of the router, and the attacker relays that information to the router after stealing the data they need from it. Using this line of code, "ettercap -T -q -i eth0 -M arp /victim ip///" with "victims IP" replaced with the IP address of the target system, we are able to launch a special type of DDOS attack. To simplify the code it in essence tells Ettercap to begin a MITM attack but instead of passing the packets along to the router, it drops them. This is known as ARP poisoning, and is a very effective was to launch a n upper scale DOS attack.

To better understand ARP Poisoning we have to first understand what an ARP Table is. An ARP table is used to manage connections between devices and services on a network by storing correlated IP and MAC addresses. What this means is that each device connecting to the system is identified and routed through this tables functions. The attack is carried out by tricking this table into incorrectly sending the sensitive transition data to the wrong location. It is almost comparable to changing the mailing address on a delivery to steal someone's package. It does this by waiting for the table to send out ARP request packets these packets are a wide broadcast call on the network asking "Who belongs to which IP and MAC address?" Upon receiving this request the attacker will send back a spoofed response packet. This packet lies to the ARP table to convince it to assign the attacker to a position they are not supposed to be in. There are many different way an attacker can reassign themselves in the table but for our purposes we will be focusing on reassigning ourselves as the router. In doing so the network will send all traffic through an attacker's machine, allowing them to delete, steal, change, or add data as they see fit.

Once an attacker is assigned to the appropriate spot in the ARP table there is a wide range of attacks that are easily executed. An attacker can simply sit quietly avoiding detection and skim valuable information off the network's traffic. Another choice is to use the response packets that it is sending back to the network to inject hostel code into any number of the networks systems. Because the network already sees the attacker as part of the network there are normally very few, if any, defenses set up to stop these kind of attacks. This is why ARP poisoning is such an effective DOS attack. Unlike standard DOS attacks, attackers don't have to overpower the victim's system, this attack can be done to a powerful target from weaker attacker. Not to mention it is significantly harder to detect. Instead of flooding the systems logs with requests that may trigger DOS defense protocols, it simply takes the

packets it's receiving as the systems "router" and drops them. Because of this, no defensive log is created, allowing the attacker to stay undetected for longer. In his article detailing hostel JavaScript injections using ARP exploits Dave Ahmad (2009) says "Although both ARP poisoning attacks and defenses have been known for years, countermeasure, are rarely implemented. This makes ARP poisoning an interesting attack vector that we can expect will be exploited even more in the future. Active exploitation of such attacks also shows that we must not ignore network layer vulnerabilities."

While this attack is effective and meets all technical definitions of a DOS attack, it does not really fit our "overburdening" model. Service is still being denied but instead of the service being overloaded, the information being sent and received is simply lost in transition. While this does burnen the system with fail logs, the overall denial of service is due to the loss of data. Because of this, in order to see how different variation of this attack affect a system we will also be executing standard flooding DOS attack through Ettercap. This attack will use the Ettercap Graphical tool plugin to send a burdening amount of requests to a service. Logically though, because our opponents are using systems with equivalent computing power, we will have to use multiple machines (our three Kali Linux machines) in order to encumber and in the end crash a service. Between these two attacks, launched from our three machines we should be able to target an IP address and crash all of its services at will.

Lab Systems Settings and Description

We performed our experiments in a laboratory environment inside of a virtual machine network. This virtual machine network was comprised of a Apache 2.2 server with the Ubuntu distribution, a Windows 7 machine running HoneyBOT, a Windows 7 machine running a firewall called Comodo, a pfSense firewall, a defensive VM running Security Onion 12.04, and finally a offensive VM running Kali Linux. The Security Onion and Kali Linux machines would not be the targets of the attacks in our exercise. Our main priorities in terms of protection was the Windows 7 machine that was also running HoneyBOT and the Apache 2.2 server. We will now briefly explain the basic function and setup of each of those parts.

Security Onion

Security Onion's main purpose is network security. It has three main features that it uses to give its users this. These are intrusion detection, network security monitoring, and log management. In order for Security Onion to perform, it combines the use of a few applications. In this lab environment the main applications that will be used are known as Sguil, Squert, Snorby, and finally ELSA. Sguil's main purpose is recording and capturing network traffic in the form of events, information from sessions, and then packet captures. This a visual representation of the data and information flowing through a network, which leds to proper decision making. The next application is known as Squert. Where Sguil captures the data, Squert helps present it. Basically Squert utilizes the data collected and morphs into something that is easier to read. It also allows the user to better extract the data from Sguil through queries. Then we have Snorby. Snorby, like Squert, helps extract and display information captured through other systems (such as Sguil). However, the main difference is that Snorby is a front end web application and it can also interact with systems other then Sguil, like Snort. The final application we have is called ELSA, which stands for Enterprise Log Search and Archive. When monitoring a network, the amount of data being stored can become cumbersome to search through, especially when it can reach such large quantities. That is where ELSA comes in. It helps its users search through logs so that they can much more easily find the information they need in a timely fashion (Sobol, 1997).

Kali Linux

Where Security Onion is our eyes, Kali Linux is our sword. Kali Linux is a linux distribution that is formed with penetration testing in mind. It sports a wide variety of tools that can be used in the offensive aspect. These can include tools such as the exploit search/usage of Armitage, the reconnaissance capabilities of Maltego, and many more. With regards to our laboratory exercise, we will mainly use Kali Linux's reconnaissance to map out our

potential targets and it will be our forerunner in launching offensive attacks against other teams (which we will explain in a further section).

pfSense

The next part of our setup is the shield. The pfSense application is the main foundation for our shield. It sports a wide variety of capabilities that can be used, including dynamic DNS, wireless access points, IPv4/6 support, NAT, inbound load balance, and many others. However, its most important feature is its firewall capabilities. This will be the major feature we will be taking advantage of in our lab environment. pfSense will serve as the frontline defense against the outside world to protect our information assets. We will configure the assets and the firewall in such a way to ensure that no unauthorized actions can be taken against the network from outside the firewall or from inside the firewall.

Comodo

Comodo is another part of the shield of our network. It is firewall that is focused on filtering out client/workstation based activities. It can do things such as whitelisting trusted publishers of files, controlling which applications can run, personalized prevention rules, behavior analysis, and even malware/virus protection. We will be taking advantage of all of these features to ensure our Windows 7 information asset is protected to the best of our abilities. Comodo has come under fire recently for various security issues. The two most recent ones are the GeekBuddy VNC server vulnerability and the Web Security Vulnerability. The GeekBuddy vulnerability that was brought to light a week ago automatically installs a VNC server and enables it (tav..., 703, 2016), Because of this we have chosen to remove GeekBuddy from the computer. The Web Security vulnerability that was found about a month ago was primarily dealing with the browser that Comodo provides, so as a precaution we didn't utilize the browser (tav..., 704, 2016). For precautionary measures, we will be ensuring that neither of these services are installed/used for the remainder of the exercise.

Apache

Apache is one of our main assets. It represents our web services and is the server for them to run on. In a business setting, it would be imperative to keep this operational and maintained. In the laboratory environment, each group member had an apache server which they ran and maintained. This means we have 3 total Apache servers to protect. These Apache servers are of configuration 2.2 and are running on Ubuntu linux distribution.

Windows 7/HoneyBOT

Finally, we have information assets in the form of Windows 7 work stations. Similar to the Apache setup, each group member has been assigned a Windows 7 machine. This means we have a total of 6 information assets we need to protect. Three Windows 7 machines and three Apache servers. Part of the defensive measures to protect the network is setting up HoneyBOT. HoneyBOT is a honeypot which acts as a distraction for any attackers, making them think they're attacking the real asset when in fact they're attacking a fake one. It is sort of like bait on a fishing line.

Topology

Now that we have explained the features of our network, let us discuss the topology. First, the pfSense firewall will be the main gateway into our network kingdom. It is from here that all inbound and outbound connections will pass through. As a result of this choke point, it is one of the major security checkpoints to ensure our kingdom stays safe. As a result, we have put all of our main assets behind this file and set it as the default gateway. So a total of 3 Apache 2.2 servers, 3 Windows 7 HoneyBOTs, and our Windows 7 Comodo are behind this firewall. Outside of our firewall we have our remaining two services, the Security Onion and Kali Linux. The reason that these services are not behind the firewall is because they will not be the targets of attacks. Figure 1 demonstrates a graphical topology of our network.

Third Party Resources/Applications

Malwarebytes

Along with the resources that were used above, we installed various third party programs to help aid in our efforts to protect our assets. The first program was anti-virus. We installed an antivirus on machines that we felt might be the focus of attacks. The program we used was called Malwarebytes. This program monitored the systems, such as one of our Apache servers, for suspicious activity through the use of heuristic analysis, signature based detection, and rootkit detection. Heuristic analysis is a method that revolves around finding activity or actions that aren't normal for the machine it is used on. The signature based detection method involves using a file of virus/malware signatures to find the malicious programs. Signatures are basically activity, patterns, or actions that a virus/malware is known to have or use. Finally the rootkit scan. The rootkit scan specializes in finding rootkits, which are malware designed to give the malicious user administrative permissions.

Nessus Vulnerability Scanner

The second third party program was used was Nessus Vulnerability scanner. What this program does is it will scan a target looking for potential attack vectors and vulnerabilities that can be exploited. This can be used in two main ways. The first way is to perform a recon on enemy systems. This can help narrow down ways to attack enemy teams in our lab environment and test our attack methods against them. The second way is to use it against yourselves. This lets you help identify areas where you defenses are weak. You can then use this data to build up specific defenses for specific vulnerabilities.

pfBlockerNG

The next third party program we used was something called pfBlockerNG. This is a package that can be downloaded to your pfSense firewall that can help create rules to block against attacks. Specifically, rules that target IP addresses. Some of the key features that you can do with the package is block entire countries, download IP address lists and automatically create rules against them, manage the rules, and view a live event logger. It also has an ad blocker to help reduce the impact ads have on users behind the firewall. All of these together can greatly buff up your firewall against malicious attacks. In the aspect of the other teams, there wasn't much we needed to do with pfBlockerNG. It's main purposes was experimenting with how it worked when blocking the entire enemy team's IP address ranges.

Defense

One of the initial things we did to secure our systems was to harden our systems against basic attacks. The first thing that this involved was instituting secure passwords that can resist password crack attempts. There are a variety of online applications and websites that can check the strength of your password. One such website that we used was howsecureismypassword.net. This was especially useful since one of the major attack methods that another team in our project is using is password cracking. One of our more easy passwords for a windows machine would take four hundred years to crack. So by utilizing this website, we produced our own password policy. The policy was as follows. The password should not contain any words that are easy to spell and if able; now words at all. The password must contain at least 1 character, 1 uppercase letter, 1 lowercase letter, and 1 number. With this policy, are systems were highly resilient against password cracking attacks.

The next defense activity we performed was scanning ourselves for vulnerabilities. This would help us harden our systems and provide an outline for potential attack vectors. The program we used was the same one talked about in the lab environment section, Nessus. Initially, our main purpose was just to create a log of vulnerabilities. Not harden them right away. The reasoning behind this was because we wanted to see if the enemy teams would exploit them. If they did, we would investigate how. This would help cement a better understanding for the computers and security. However, the teams never did attempt to exploit them. The only exception is what the honeyBOT picked

up. Which we will go into more detail below. But it the HoneyBOT found that one team may have been using ports 137-139, which is also what Nessus identified. So we closed those off.

DDOS attacks are one of the easiest attacks to use and one of the hardest attacks to defend against. The issue with DDOS attacks is that if it's being attempted on a system that has a lot of users then the DDOS attack is hard to defend against because you're not sure if this overwhelming traffic is genuine. When on a home network it is very easy to defend against DDOS attacks because you should be able to monitor your traffic and distinguish a DDOS attack from regular traffic. The main focus is prevention because it is impossible to stop a DDoS attack and very hard to properly respond to such an attack. The only way to respond to a DDOS attack is to check your logs and to find patterns within the ip address to understand who and where the attack came from.

Some common techniques for preventing DDOS attacks (without outside help) include using a firewall which acts as a "chokepoint", route filtering techniques which involves using a remotely triggered black hole and putting all the DDOS traffic into this "dead zone", and unicast reverse path forwarding which determines the reachability of the connection and if it fails it rejects the packets. There are two disabling techniques which involve disabling unused services to prevent attacks from using these ports and disabling IP broadcasts prevents ICMP floods.. Load balancing relies on the ISP to allow you to use more bandwidth during an attack. The last resort is to change your IP address to dissociate yourself from the attack. Although it won't take long for the attacker to notice, a honeypot can be set up to distract an attacker long enough so you can be alerted of a DDOS attack giving you enough time to respond. (Douligeris & Mitrokotsa, 2004)

The problem with monitoring all the traffic is that you can't tell if the connection is genuine or part of a DDOS attack. Another way to distinguish the traffic is to look at the IP location and see if this botnet is operating in a common location and block all IPs from that location, although it can affect the availability of your network. An issue that arises is that many companies don't want to pay somebody to monitor the traffic because DDOS attacks don't happen often. A tool that is good for monitoring packet traffic is Wireshark. It runs on all operating systems and provides a lot of tools that can give you detailed reports and analysis of traffic. This is ideal for home networks because there isn't a lot of genuine traffic to be monitored, making a DDOS attack easy to spot. Theoretically speaking, the best way to defend against a DDOS attack is to keep up with the attack and keep blocking if you're fast enough.

If your business is unable to have an in-house security division then there are two options. Relying on multiple ISPs and using cloud mitigation services. Relying on multiple ISPs allows you to have a backup if one of the services your ISP provided is currently getting attacked which would allow you to mitigate all of this bad traffic to one of their servers dedicated to DDOS mitigation. The problem with this is that the ISPs don't communicate with each other which could be a problem if both networks are being affected by DDOS attacks.

According to Nirav Shah(2014) brings up the point that there is no firewall protection against DDOS attacks as it causes a bottleneck in your overall performance. The nature of DDOS attacks involves disguising the traffic as wanted traffic in the ports that have to be open to prevent it from being blocked. Having all the traffic checked causes a major availability issue as it will take longer to receive packets. Malwarebytes is a free scanner tool that also blocks incoming packets if it doesn't recognize the source. When you first start it, it blocks all of your application ports to make sure they aren't malicious and to confirm with you that you want that port open. This is a great tool for the regular user who doesn't know much about IT security and it's also free.

The only real defense against DDOS is to prevent it from happening. There are many companies that are making money off this situation by providing products for DDOS attacks such as 3rd party mitigation services and scanners. This option is very expensive but is definitely the best protection. Letting a 3rd party deal with mitigations requires no security professionals to be on site of the network and the traffic gets mitigated as soon as it's detected. Radware has an IPS called the "DefensePro" that has a mitigation device but is very expensive. They also offer the service of attack mitigation that allows them to mitigate traffic once you have notified them of an attack, but it's best to use this paired with DefensePro so you have an IPS on site but this option is ideal if you don't have the capital for a DefensePro but want some sort of intrusion response.

One of the only things a firewall can do is detect and block incoming traffic during a DDOS attack. Unless it's one system commencing a DoS, it will be very hard to detect the zombie bot IPs that are attacking your system. You could block a range of IPs but you might block genuine traffic that will affect the availability of your network.

A big problem arises when it comes to ARP poisoning due to the fact the ARP tables caches any ARP reply that is sent which leaves your system open to more exploits. There are some easy solutions such as using ARPWATCH or using permanent entries for IP/MAC address pairings. ARPWATCH is a monitoring program which scans and logs known IP/MAC pairings and notifies the user when there is a change. According to Nathaniel Eisele, these methods are "incomplete" in terms of their features. With ARPWATCH, you can't dynamically assign IP/MAC addresses and with permanent entries you can't detect destination failures or dynamically assign addresses. One of the major benefits of ARPWATCH is the fact that it records all activity which could lead to the identity of attackers but due to the nature of ARPWATCH, it requires someone to actively look at the monitor. ARPWATCH is great if you have the resources to maintain it. Businesses that operate internally tend to block all other MAC addresses to prevent outsiders from connecting to the network as a precaution if they are unable to afford the equipment.

There are many precautions that can be taken to secure your network from malicious attacks. One of the first things that should be done is to use a program like OutPost Firewall Pro that denies ARP scans which should prevent your network/system from coming up when attackers use a network scan program. It's completely unnecessary to allow your personal system to be scanned but if you're running a large network, it would not be advised. You can also divide your large network into subnets which would prevent other systems from being affected by attacks, you can permanently assign addresses and choose who you communicate with, you can setup an alarm for ARP replies, and you can add static MAC addresses but this method only works if there are no mobile devices. Many firewalls already contain a defense that protects the ARP cache which prevents attacks using the ARP table.Ironically, the same company that created a popular ARP poisoning attack program, Netcut created by arcai, created a defense against the same program and it is called Netcut defender. The attack program itself, Netcut, also has a feature to defend against arp poison attacks but some people are not willing to download a program that may be considered illegal when used.

According to Mohamed Gouda, one of the answers is to add two new protocols that involve using Invite-accept and Request-reply. Invite-accept has computers communicating their addresses on connection to a secure server and request-reply allows each computer to resolve their address on a network. Both of these protocols involve storing variables and comparing the active variables with the stored variables to confirm an address. The problem with using protocols is the fact that it relies on the secure server, which can be protected if a backup server is added and reads all the information being sent to the main server. This allows the backup server to be ready at any time if the main server is taken down. Another solution is to add an authentication mechanism to the operation code on the ARP header. This can be done by adding values 5-10 to the field. One of the first values that are added is host registration, which stores bytes that are sent from a user. This allows the system to detect if any connections being spoofed, which allows the system to discard the spoofed connection. This method is protected by sniffing because the bytes are broken up.(Gouda, 2003)

Some solutions offered by Cisco involve using Dynamic ARP Inspection(DAI) and Dynamic Host Configuration Protocol (DHCP) snooping. DAI involves comparing MAC and IP bindings received to a trusted bindings table. DHCP is required because it populates the trusted binding table. In the event that DHCP isn't be relied on, DAI can statically assign IP addresses. DAI can also be set up to drop packets that are coming from spoofed addresses or invalid IP addresses. DAI has a feature that allows you set the maximum rate of incoming ARP packets, this feature protects you from a DDoS attack as well. If the maximum rate of packets is reached, it will deny everything until the user intervenes or until the timeout is reached. When DAI is running, the network is setup so all ports connected to the switch are trusted, while all the other ports are untrusted. In addition to the list of binds created by DHCP snooping, DAI has an ARP access control list(ACL) table(configured by the user) that has authority over the DHCP snooping bind table.DAI also has a feature that drops any packet that has mismatched addresses in the header and body which is useful against spoofing.DHCP spoofing is only an option if the DHCP ports are open(ports 67(King)

Network Defense Events

Through the course of the project, we were tasked with defending our systems against the other teams. The other teams utilized two main forms of attacks. One attempted to exploit password cracking to gain access to our systems while the other specializes in writing and installing viruses to cause havoc on our systems. Throughout the course of the year, we had minimal incidents happen to us. We started the attack phase of the project with changing all of our passwords. These password followed the criteria of including lowercase, uppercase, special characters, and numbers. These increased their ability to resist the password cracking attempts to a high degree. Following this, we set up our Security Onion and HoneyBOTs to conduct monitoring exercises on our network. We used the data from these to help identify potential threats or breaches.

Defending with the Security Onion

We experienced issues in getting our Security Onion to work. What we found was happening was that the Security Onion was monitoring the correct adapters, but for some reason wasn't picking up activity from our machines. The only time it would pick up activity from those machines is if we used those machines to target the security onion directly. We attempted to debug it through different installation/setup parameters, but none of them produced results. Two of the ways we were able to tell if the Security Onion was operating properly was by forming two rules for the monitoring software. One rule was to track any and all ICMP protocol packets going through our system. The Internet Control Message Protocol (ICMP) is mainly used when performing ping commands on the command prompt in Windows. So we would ping different locations with the hopes that the Security Onion would retrieve them. Sadly, it did not. It was tracking ICMP request, but their sources and destinations were unknown. Our second attempt was writing a new rule to track the ICMP protocol packets coming from a specific destination, one of our Windows 7 machines. It would only track these when the Windows 7 machine pings the Security Onion directly. After attempting to debug the Security Onion for 21 hours total and the time we had on the project dwindling, we decided to abandon attempts to get it working correctly. However, one defense mechanism worked. The HoneyBOT.

Defending with the HoneyBOT

In our HoneyBOT we were getting logs of activity on the network. While it wasn't picking up data from the team attempting to password crack, we were tracking the virus's team activity. While the virus team was testing their attack methods on their own networks/machines, we were able to see how their attack works. We started finding the IP addresses they were using and the main ports they were using for the attacks. We saw spikes in activity in the ports 137-139 while they were testing. These ports are used for the netBIOs and are common attack vectors. We first were surprised that HoneyBOT was somehow monitoring their activity that was behind a firewall. It highlighted the importance of properly designing your network, because the last thing you want is someone performing network monitoring on your network without your knowledge. It was because of this flaw in their defense that we were able to see their activity and start formulating defenses.

Defending with pfSense/Windows Firewall

After witnessing the attacks on HoneyBOT, we went into our pfSense firewall and wrote a few new rules for this team. First, we closed down any and all activity through the ports 137-139. These ports are commonly closed for network defense and was also picked up by Nessus. As a security measure, we also instituted new rules in the Windows based firewalls to block these ports. The next thing we did was take the IP addresses that the pfSense firewall picked up and blocked each and every one of them. We were able to verify this was successfully stopping the enemy virus team when they attempted to ping one of our machines and the requests were dropped. With these defenses up, we were confident in our abilities to prevent attacks from the virus team. However, we now had to deal with the password cracking team.

The Suspected Breach

On one of the final weeks, we had a suspected breach of our network. On one of our machines, the password was somehow reset to the default password that was there at the start of the year. Finally, the machine also had its network adapter settings reset. This meant it was no longer behind the firewall. Finally, the account we had access to has had its permissions set to guest. This meant we no longer had control of the computer. We started going over the

logs in HoneyBOT and didn't see anything suspicious or activity from either of the enemy teams. This machine, however, has had a breach in the past. Towards the middle of the project, someone had left the machine logged in over the virtual server. The password cracking team saw this and had utilized it to their advantage. While we only know they changed the login picture, we were unsure of the full extent of the damage on the machine. With our guest permissions, we attempted to look at any and all logs that we had access to on the machine. Sadly, we realize that without administration permissions; finding and fixing the issue would be problematic. We decided that the only chance of salvaging the situation was hacking our own machine to regain access to it. We exploited the sticky keys/startup repair vulnerability that all Windows 7 machines have. This had been our secret weapon, if we ever were in a situation that needed it. With this vulnerability, we were able regain control of the machine.

Through more investigation into the system and trying to decided how it was compromised, we came up with the following results. First, we don't think it was something based in a case of malicious software. We say this for a few reasons. First, we didn't see anything in the logs of the machine itself (along with the HoneyBOT spoke above). We also checked the network monitor on the system and didn't see any abnormal activity. Following this, we checked the process/task manager to look for potential programs running. We couldn't find anything. We then inquired to another team to see if they were the ones behind the reset. What we found was that they were not, but were encountering an issue similar to ours. They kept losing control of their pfSense firewall, despite resecuring it. Both teams worked together to figure out how the remaining team was able to gain control of the systems. Another investigation was launched on the other teams pfSense firewall and came up with the same results we did with our Windows machine. We concluded that they must be regaining control of the system through the virtual machine system itself, not the actual computer. That's when we realized their attack method. The team had set a virtual machine checkpoint to the point where it was already logged in. This meant that whenever they wanted to take control of the system, they just restored to that checkpoint. This activity does not appear in the logs we had access too. The restore button and checkpoint button are also not protected by a password or any security. They were able to restore the checkpoint without even need to log into the machine. Which is normal for the Hyper-V virtual interface. Finally, there is no way to block this command from executing. So the only solution was to simply create a new checkpoint that was at a point in time where it was secure and not logged in. Once this was done, we no longer had issues with these machines. We then fixed the issues, set the password, and installed an antivirus as a precautionary measure. We were not able to find anything on the machine still. So we decided that we needed to attack the password cracking team directly, to hopefully cut off their offensive advance at the head. This would also help serve as a distraction for them.

The Counter Attack

After the potential breach incident, we performed the vulnerability spoke about above. The basic sense of the vulnerability is that the system restore on failed startup for windows machines allows you to open notepad to view a privacy policy. You can then browse the computer's files. From here, we went to the System32 folder and found the executable that is performed when you push shift 5 times and attempt to enable sticky keys. We renamed this to something random, then named the command terminal the same thing that the executable was named. We then restarted the machine. Once restarted, we went to the login screen to get into the computer. From here, all you have to do is push shift five times. This will then start the command terminal with administration permissions. From the command terminal, we then made our own administration account named "Moderator" with a secure password. From there, we took away the administration permissions of the account the password cracking team was using. This took away control of the machine from the other team and gave us full control.

Now, we didn't want it to be immediately obvious we had taken control of their system. The main purpose of this counter attack was to serve as a distraction. So the first thing we did is to change the files that renamed back to what they are normally. We then went into the local security policy and made it so that the login screen will not save the last login. This means when we log out, the other users will have no idea we were logged in. Finally, we hardened the enemy's systems against the exploit we used. This means that if they attempted to perform the same exploit to regain control of their system, they would be unable to. We also hardened our own systems against this exploit at the time too. So the enemy team has no ability to see that we had administration permissions to their account, they have no ability to regain control of their machine without the professor's help, and finally they don't have administration permissions on their own machines. We performed this exploit on all of their machines that were in use, except two.

This is to allow them to still complete the project. The final act in our counter attack was giving access to these machines to the virus team. This meant the virus team could install their payload on the password cracking team, thereby distracting both teams. While this was going on, my team continued to harden their systems and launch the DDOS and ARP poison attacks; which we will go into next.

Conclusion

With all this in mind, we can quickly see that DDOS and ARP poison attacks are becoming a very serious threats in today's cyber world. With our virtual environment laboratory setup, we can see that even a bare minimum, highly protected network is still vulnerable to this attack. We also saw with our virtual environment how to deal with potential security breaches in an account along with defending our network against further attacks. By performing a vulnerability scan, we were able to narrow down potential areas that malicious attackers could use to compromise our systems. We acquired malicious IP addresses and created firewall rules to block them in our pfSense firewall. Our HoneyBOT was the main defense tool that detected malicious network activity. It was able to identify which IP addresses team 2 was using and which ports they were using over the network. With this data, we blocked both their IP address and the port numbers that were being used (137-139). We further reinforced our defenses by also blocking those ports on the Windows based firewalls. We found that one team was exploiting vulnerabilities in the setup of the Hyper-V virtual machine environment. Using the flaws in not automatically logging out machines when turning off to gain access to systems if a user didn't' log off of them. They also set system restore points to vulnerable periods and restored to those periods in order to continue their attacks. To combat these techniques, we tried to institute strategies to accommodate for them. This included double checking to ensure all machines were logged out. We also ensured that if any restore points were set, we would reconfigure them so that they were at points where it was secure. Following these defense measures, we performed a series of counter attacks. The first one being the DDOS attacks. The second attack we utilized was the sticky key exploit to gain administrative access to their windows machine.

By examining the threat type earlier in the article, we found that DDOS attacks specialize in taking down servers and are often hard to counter. Using our Kali Linux machines, we were able to explore using this type of attack against other teams. Of course one of the most important things is learning to counter it, which we covered in the defense section. It was here that we found methods of utilizing firewall routing rules and "black holes" can help counter DDOS attacks. However, the biggest issue with this system is differentiating between genuine traffic and malicious traffic. Having a firewall chokepoint can also negatively impact your organization's ability to offer services to clients, depending on the situation. So with this information in mind, we found that the only viable solution to this attack is attempting to prevent it. This includes having your network be able to handle large amounts of traffic at once and/or having third party mitigations. When we compared this solution to our own virtual environment, we found it would simply be unfeasible. Our network is too small to handle large loads of traffic at once and our resources in the real world are too small to be able to afford third party mitigation. This can translate into real world scenarios where our small network can represent the many small companies in today's business environment and how susceptible they are to DDOS attacks.

For ARP poison attacks we found that similar results. They are an attack that is easy to perform, but hard to counter. To counter it, we recommend investigate invite-accept and request-reply style protocols and implement them into your network. Along with this, taking preventive measures can help stop attacks before they start or do a significant amount of damage. Instituting a monitoring program like ARPWATCH to help keep track of MAC and IP addresses and to create alerts when a change is attempted. Prevent a reconnaissance of your ARP tables can also help you greatly. Finally, putting in analytical practices into hardware, like DHCP spoofing, can prevent poison attacks. We also recommend looking into third party programs or packages for your various network defenses. These can help further reinforce your defenses. Both your frontline and backline defense mechanisms. Performing attacks or recon against yourself can often go a long way in defending against attacks. Using a program like Nessus to find weak points or vulnerabilities in your system can allow you to purge your network of vulnerabilities before a malicious user takes advantage of them. Along with this, having a secure password policy can help prevent any malicious actions that center around password cracking. Utilizing websites or programs that will determine how "good" your passwords are can help you form your password policy.

With the results spoke about above, we found how hard it was to protect against DDOS and ARP attacks. We discovered new ways to defend our systems from a variety of attacks. The recommendations we have given based on these learning experiences should provide readers sufficient knowledge to begin protecting themselves and others from these attacks.

Work Cited

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. Computer Networks, 44(5), 643-666. doi:http://dx.doi.org/10.1016/j.comnet.2003.10.003

Gouda, M. G., & Huang, C. (2003). A secure address resolution protocol. *Computer Networks,* (41), 57-71. Retrieved April 10, 2016, from https://www.cs.utexas.edu/~gouda/papers/journal/JA2.pdf.

Gupta, S. (2012). *Logging and monitoring to detect network intrusions and compliance violations in the environment.* (Research Paper). SANS Institute Reading Room: The SANS Institute.

K.R, V., & Gudar, B. (2014). Detection of ARP Spoofing Attacks Using ICMP Protocol. *International Journal of Computer Science and Mobile Computing, 3*(5), 1055-1060. Retrieved April 10, 2016, from http://www.academia.edu/7201783/Detection_of_ARP_Spoofing_Attack_Using_ICMP_Protocol_

King, J., & Lauerman, K. (2014, January 7). ARP Poisoning Attack and Mitigation Techniques. Retrieved April 10, 2016, from http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-6500-series-switches/white_paper_c11_603839.html

Leach, S. (2013). Four ways to defend against DDoS attacks. Retrieved from *http:*//www.networkworld.com/article/2170051/tech-primers/tech-primers-four-ways-to-defend-against-ddos-attacks.html

Mittal, Akash, Ajit Kumar Shrivastava, and Manish Manoria., 2016, "A Review of DDOS Attack and Its Countermeasures in TCP Based Networks." *International Journal of Computer Science & Engineering Survey IJCSES* 2.4 (2011): 177-87. http://airccse.org/journal/ijcses/papers/1111ijcses13.pdf

Shah, N. (2014). The DDos myth about the firewall and IPS. Retrieved from https://www.corero.com/blog/609-the-ddos-myth-about-the-firewall-and-the-ips.html

Sharma, D., Ms, Khan, O., Mr, Aggarwal, K., Ms, & Vaidya, P., Ms. (2013). A New Approach to Prevent ARP Spoofing. *International Journal of Innovative Technology and Exploring Engineering, 3*(1), 80-82. Retrieved April 10, 2016, from http://www.ijitee.org/attachments/File/v3i1/A0905063113.pdf

Sobol, M. I. (1997). Firewalls. *Information Systems Security, 6*(1), 20-28. doi:10.1080/10658989709342525

tav... (2016). Issue 703: comodo: Comodo internet security installs and starts a VNC server by default Retrieved from https://code.google.com/p/google-security-research/issues/detail?id=703

tav... (2016). Issue 704: comodo: Comodo "chromodo" browser disables same origin policy, effectively turning off web security. Retrieved from https://code.google.com/p/google-security-research/issues/detail?id=704