Global Information Assurance Certification Paper

Success Rates for Client Side Vulnerabilities

GIAC (GCIH) Gold Certification

Author: Jonathan Risto, jonathan.risto@hotmail.com

Advisor: Manuel Humberto Santander Pelaez
Accepted: March 21, 2016

Abstract

The user is the weakest link in the computer security chain. From clicking on links that they shouldn't to having weak passwords, it generally comes down to the end user doing something they shouldn't. If the user runs a piece of malware or opens an infected file, will it always lead to a compromise? This paper plans to test if client-side exploits will always function or if there are additional factors to consider when dealing with these vulnerabilities and associated exploits. Is the Common Vulnerability Scoring System (CVSS) score enough to determine if a particular vulnerability is more critical than another and should be remediated sooner than another? This testing will be accomplished through the use of freely available exploitation software (e.g. Social Engineering Toolkit, Metasploit) in a closed testing environment.

1. Introduction

Networks suffer from breaches every day. Hackers gain access to environments through multiple means, but one of the most common ways for numerous years has been the user. In 2015, Verizon reported that nearly 90% of the incidents have a single common factor, which is people (Verizon, 2015). This percentage does include insider misuse and administrator errors, but malware installation by users is one of the largest problems facing security professionals today. Taking advantage of human curiosity or greed to have a user open a particular email or click on a link is a powerful and a very efficient strategy used today. At the end of 2014, Palo Alto Networks findings on malware distribution channels were that 87% of the malware was distributed/activated through email (Palo Alto Networks, 2014), a strategy which automatically involves the user.

These email attacks are targeting large and small organizations. For targeted phishing attacks[1], called spear phishing[2], 83%, 63% and 45% of large, medium and small enterprises respectively were targets in 2014 (Symantec, 2015). The attacks also happen very quickly, as Verizon reported that the median time to first click in a phishing campaign was a mere one minute 22 seconds after the emails were sent, with over 50% clicking occurring within the first hour (Verizon, 2015).

With this short amount of a time to defend the network in real time, incident responders and security professionals need assistance in prioritizing which vulnerability should be patched and guarded against first. CVSS scores do not always agree in the criticality of vulnerability, based on experimental testing of server-side exploits (Dondo,

[1] A phishing attack is an email attack that is sent to a large number of people, hoping that some people will perform the actions contained within.

[2] A spear phishing attack is an email sent to a specific individual or group within a company and is directed to that group for a specific purpose.

Author Name, email@address

Risto, & Sawilla, 2014). Part of the reason for this is that certain aspects of the CVSS score are poorly defined and often not used at all (Bhatt, Horne, & Rao, 2011).

The goal of this paper is to show, based on the testing performed, if defenders require additional information to help prioritize the remediation of client-side vulnerabilities.

2. Users – The Weakest Link

In its 2015 Data Breach Report, Verizon noted that 23% of users opened a phishing email received, and 11% clicked on the attachments (Verizon, 2015).

Topics used in phishing attacks typically link to items in the news or various celebrities (Verizon, 2015). In 2014, 71% of the phishing attacks reported by Symantec were related to financial institutions (Deschatres, 2014).

During the nine months ending September 30th, 2015 the Anti Phishing Working Group, APWG, received over 1 million unique phishing reports and over 630,000 unique phishing sites (APWG, 2015).

What do all of the statistics mean? Why does it matter if multiple thousands of emails coming monthly to the organization are spam? The reason that it does matter is that by sheer volume, some of the emails are bound to make it through to some of the end users. According to Kaspersky, just over 66% of emails were spam in Q1 2014 (Kaspersky Lab, 2014). Our protection measures cannot all be perfect.

Let's assume that a hypothetical organization receives 100,000 emails per month. Using the Kaspersky numbers, 66,000 of these would be spam messages. Assuming a spam filter that is 99% efficient, 660 emails would make it through the spam filters and end up in end user mailboxes. Given that end users are quite smart, and would realize 90% of the time that these emails should be deleted, that leaves us with 66 emails that would be opened. If we were to use the 2015 Verizon Data Breach Report number of 77% of users would open the emails, this figure would increase to 152 emails that were opened.

66 or even 152 problem emails doesn't seem like much out of 100,000 does it? The issue is that each and every one of these emails could end up as a security incident.

Author Name, email@address

If even only 2% of these contained very malicious and sophisticated malware, each and every month there are at least 2-3 serious security incidents. The 2% estimate is very conservative, in the opinion of the author, as this equates to only 0.0000454% of all spam emails being sophisticated, and that is a rather small percentage. And these are only incidents that come to the organization through spam. Add in events from inadvertent downloads, infected websites, and other more targeted attacks to the organization; there are a lot of ways that engage the end user for client side attacks each month. There is a need to prioritize the remediation of these vulnerabilities.

3. The Experiment

In order to determine how effective client-side vulnerabilities are, a method to test various vulnerabilities was devised. By testing the effectiveness of various exploits available, it will show if additional information should be considered by the defending organizations to help prioritize the remediation efforts.

3.1. Setup and Testing Equipment

Testing of the exploits was conducted within a virtual machine environment, using VMware Workstation version 9.0.2. Within that environment, the target system was chosen to be Windows 7, as it still has over half of the market share for desktop operating systems (Net Applications, 2016).

For the experiment, it was created to minimize outside factors affecting the success rates of the vulnerabilities, such as network packet loss, virus scanning, and any social factors. If the malicious file was present, it was executed for example. Both computers were created in new virtual machines and installed from ISO images. The network connections were provided by the networking ability of the VMware workstation. A new private network was created using the 192.168.1.0/24 network space. The network configuration is shown in Figure 1:

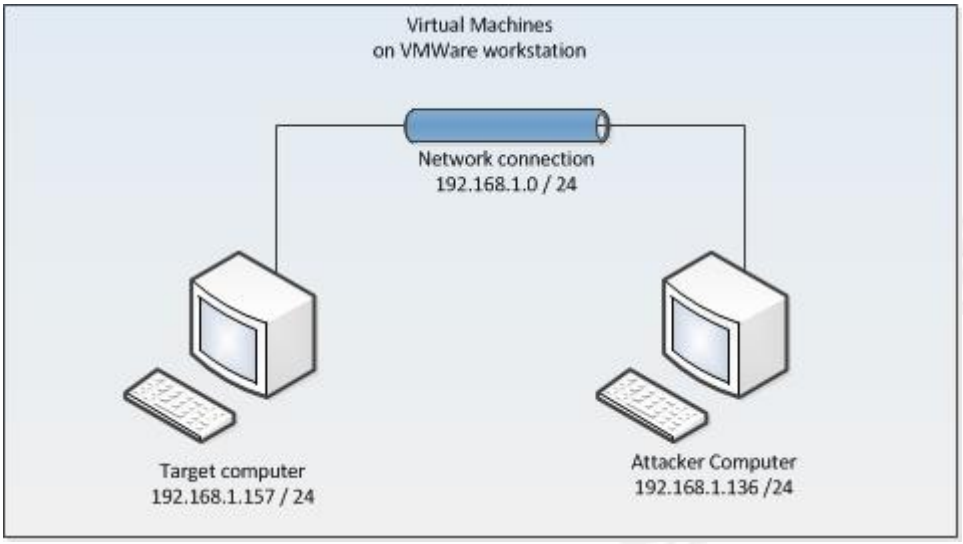

Figure 1: Network testing setup and configuration

3.1.1. Attacker Computer Configuration

The attacker machine was installed with Ubuntu 14.04 Linux distribution, and it was updated to most recent packages. After the base operating system was installed, Metasploit version 4.11.10 was installed, as was the Social Engineering Toolkit[3], version 6.5.9. With both of these attack tools, any dependent software packages were also installed on the attacker system.

3.1.2. Target Computer Configuration

The target computer was installed with Windows 7, build 7601, Service Pack 1, with no further updates performed on this system. This was to permit a modern installation of the operating system, but also, allow the use of some older software, such as Internet Explorer version 8.0.7601. Additional software that was installed on the target computer was Java, version 6u45. This release of Java was chosen to enable the Java attack used, as described in Section 3.2, to function. It was also released in a similar timeframe to the Windows 7 version used in the testing. Adobe Acrobat version 8 was

[3] The Social Engineering Toolkit is available from https://www.trustedsec.com/social-engineer-toolkit/

also installed on the target computer, to provide the software needed for testing the attack described in Section 3.2.

3.2. Vulnerabilities and Exploits Used

Client-side exploits are numerous and varied in their format. Everything from email attachments that contain Java, executables, MS Office documents, and links to downloaders are just a handful of the many different possibilities. Of the numerous options available, the following five (5) were chosen for testing as a representative sample:

1) Malicious executable created by Metasploit using the Social Engineering Toolkit (SET) and run on the target system;
2) A Java exploit set up by the Social Engineering Toolkit and run by visiting a website;
3) An Internet Explorer exploit, triggered by visiting a Metasploit-created web page exploiting vulnerability MS11-003;
4) An Adobe Acrobat vulnerability, exploiting vulnerability CVE 2010-1240 created by Metasploit; and
5) A Firefox exploit, utilizing the bootstrap functionality and created by Metasploit.

3.3. Testing Procedures

Testing of the vulnerability and exploit pairs was done by running the exploit on the target system 20 times. This execution was performed without any rebooting or changing the target system or the attacker system. By leaving both these systems untouched, it will permit any potential problems from the exploit to remain in memory, as they would under normal operational circumstances. Accurate testing information relating to each particular test is outlined in the following sections.

3.3.1. Malicious Executable

To create the malicious executable, the Social Engineering Toolkit (SET) was used. The program was executed on the attacker system, and it displays the screen shown in Figure 2. Option number 4, which is to create a payload and listener, was chosen.

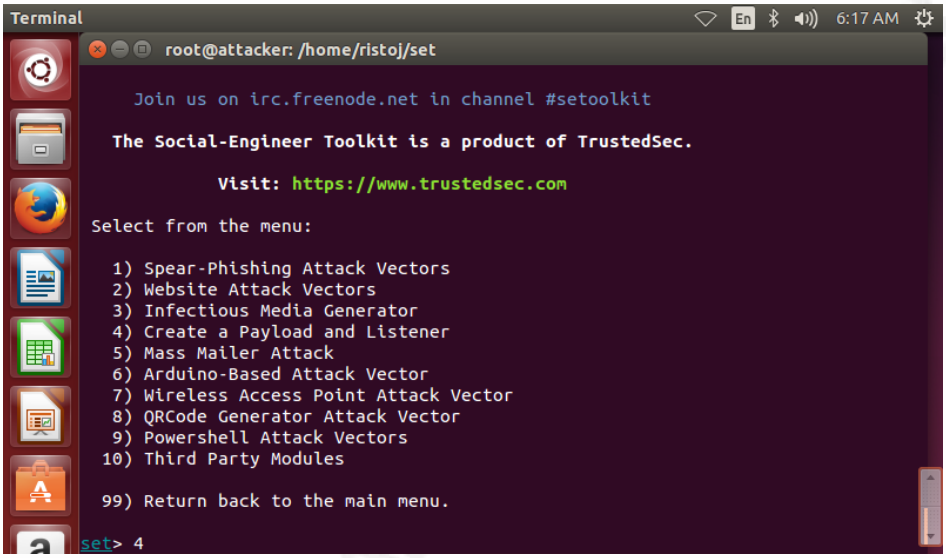

Figure 2: Malicious executable option in SET

When option 4 is chosen, a screen similar to Figure 3 is displayed. Option 1, a Windows Shell Reverse_TCP, was then selected.

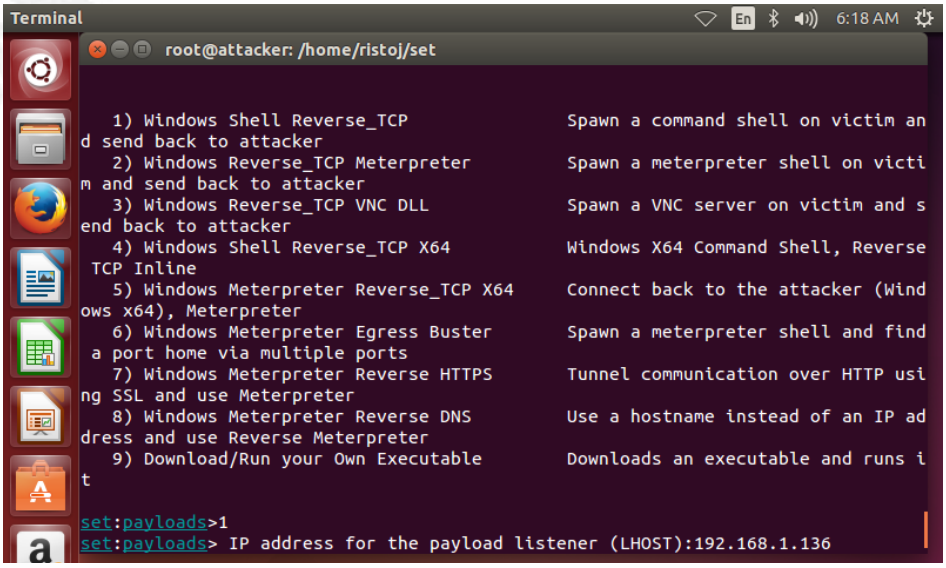

Figure 3: SET payload options

After choosing the type of payload, machine specific information for the attackers system needs to be entered. The IP address and the port on which the attacker's computer is listening for connections need to be configured. In the case of the test run, this was IP address 192.168.1.136 and port 443 as shown in Figure 4. After entering this information, SET creates the malicious executable and indicates where it has stored this information. After this, SET asks if the attacker wishes to start the payload and listener now. When entering yes, the system automatically starts Metasploit.

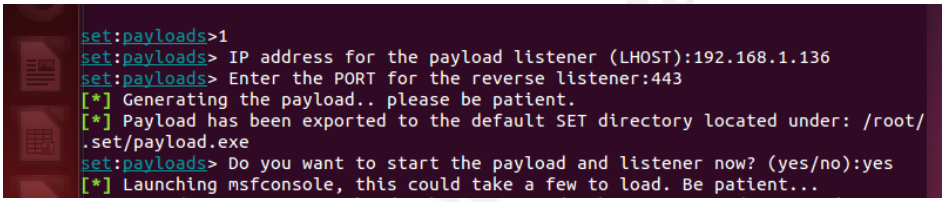

Figure 4: Final SET configuration for malicious payload

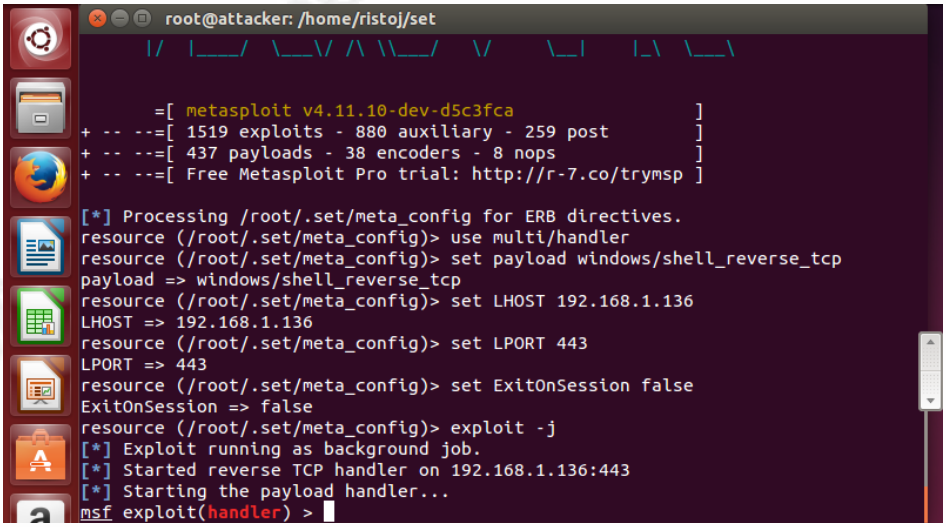

Figure 5: Successful launching of Metasploit with configuration parameters

After launching Metasploit, the system is scripted to run within Metasploit the settings as chosen, with the information that was entered. As shown in Figure 5, the system configured the IP address, port and started the reverse shell payload in the

background set with this information. Now the attacker's computer is ready for someone to click on the executable that was created.

In the testing performed, the executable was placed on a USB stick and copied to the target system desktop, as shown in Figure 6. Once there, the executable was run by double clicking it.

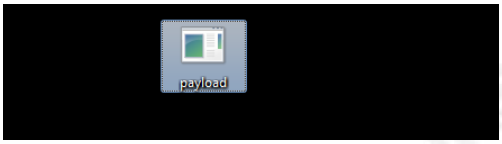

Figure 6: Payload executable as seen on target computer

When the executable was run, there were no visible programs that ran or anything that was shown to the end user. It just appears that the program didn't run, or that it was not double clicked correctly. However, on the attacker side, we can see that the program did run accurately, as displayed in Figure 7, as it connected to the attacker system. The attacker system successfully opened a session on the target computer. Using the Metasploit command "sessions –i 1", the Windows shell session was accessed.

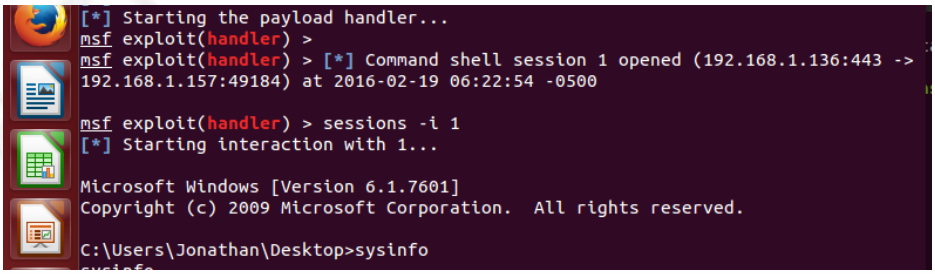

Figure 7: Successful run of the payload.exe file

For the testing, the majority of the above steps did not need to be performed. With the payload on the target system and the listened setup on the attacker system, the only item requiring completion was executing the payload to initiate the connection. For this work, the payload was executed 20 times, and the success or failure recorded for each. The results of this are discussed in Section 3.4.

3.3.2. Java Exploit

Similar to the malicious executable, SET was used to create the Java exploit. To accomplish this, option 2, Website Attack Vectors, was chosen from the SET main menu as shown in Figure 8.

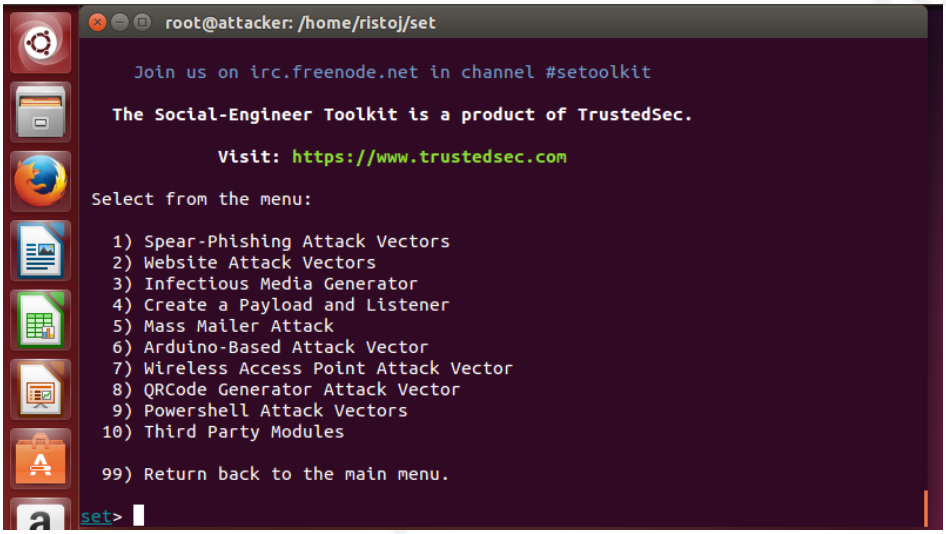

Figure 8: SET configuration for Java exploit

After choosing the attack vector, the options shown in Figure 8 are displayed. From these available options, option 1, Java Website Attack method, was adopted.

Figure 9: Web attack options in SET

After choosing the Java applet option, you are presented with options on how to create the website for the attack, as outlined in Figure 10. Options include web templates built into SET, cloning an existing website, or importing a custom homepage. For the testing work performed, option 1, web templates, was chosen.

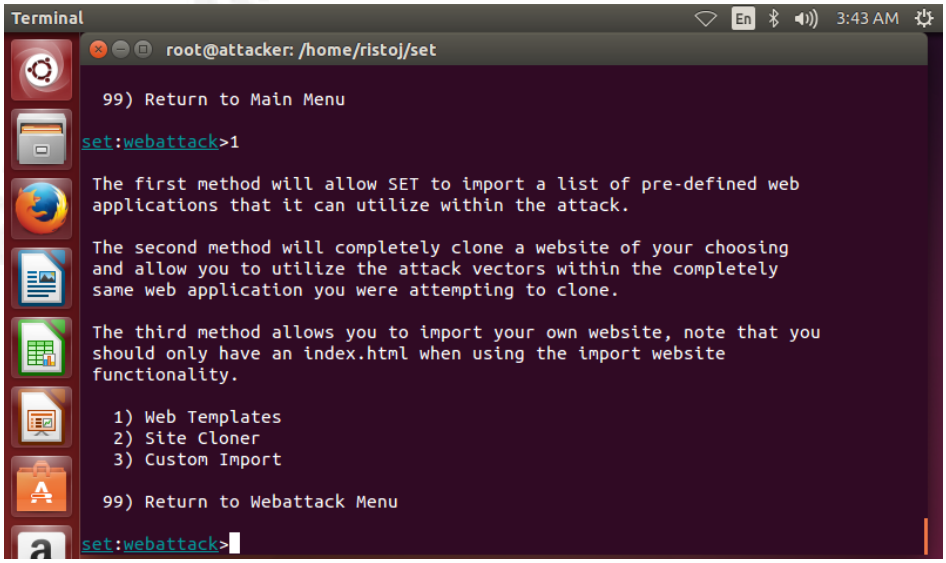

Figure 10: Options for website creation in SET

After choosing to use the web templates included within SET, the options are presented to determine how the Java applet is to be built. The attacker IP address is then entered. The next option is choosing the applet to use for the attack. For the testing performed, the applet within SET was used, by selecting option 2, as shown in Figure 11.

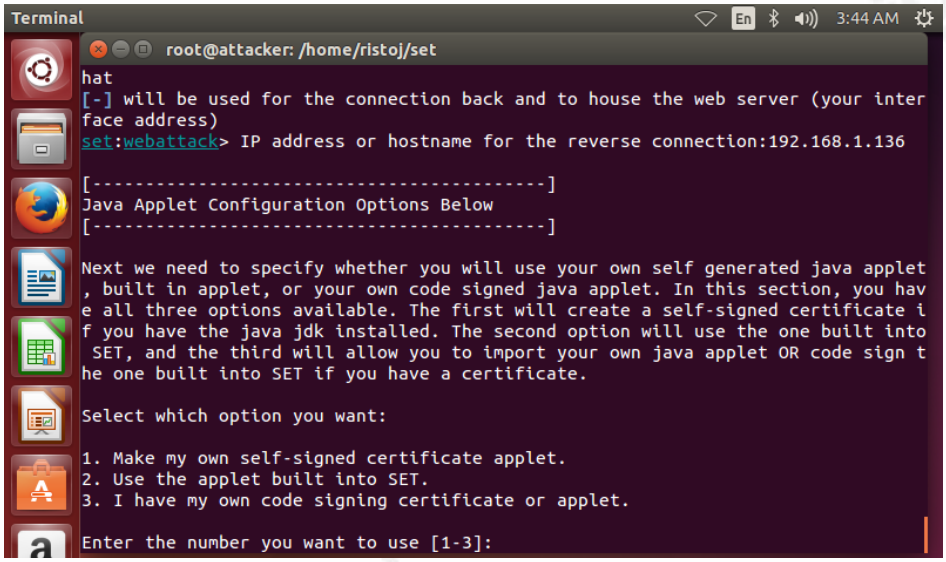

Figure 11: Options for certificates for Java attack

The following choice is needed to determine which of the built-in web templates are to be used. There are numerous options here to affect how the website will look, as shown in Figure 12, but for the testing performed, this choice did not matter. Option 1, Java Required, was chosen.

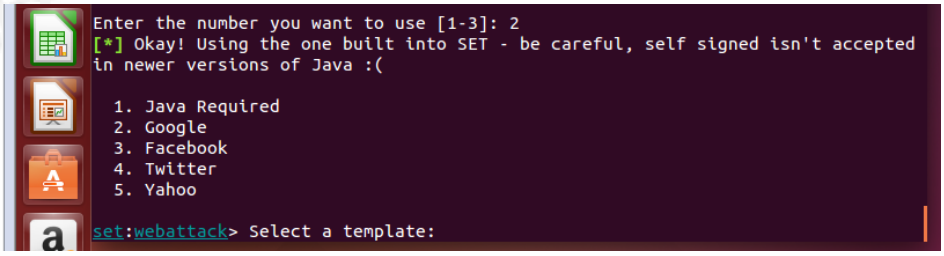

Figure 12: SET options for website creation

The following group of options permits the configuring of the malicious Java applet and what this applet will do for the attacker. In Figure 13, options are presented for the payload generation. Option 1, the default Meterpreter Memory injection, was chosen

for the testing. The port on which the listener will be configured to run is set next. Port 443 was selected for the testing.

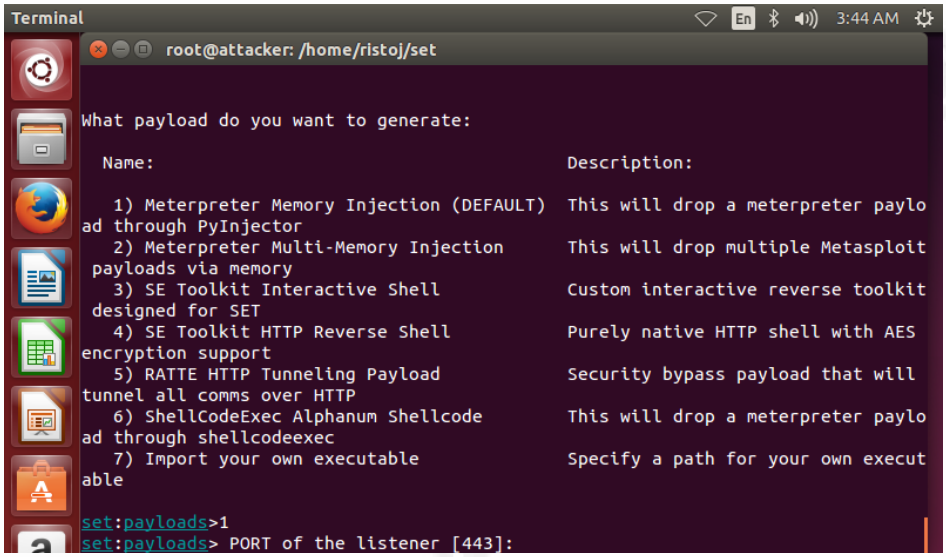

Figure 13: Payload options for website JAVA attack

After selecting the listener port, port 443 in the testing, a selection of possible payloads is displayed. Figure 14 shows the possible options, with Option 1, a Meterpreter Reverse TCP payload, chosen for the testing.

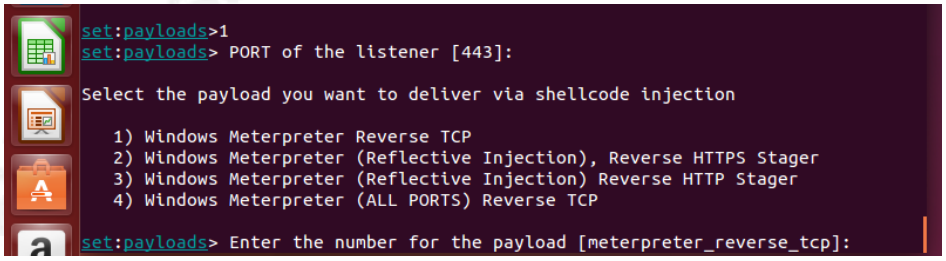

Figure 14: More payload options for Java website attack

After the selections have been made, SET creates the needed files, places them into the correct directory and starts the web applications server. Also, similarly to the malicious executable testing, SET launches and configures Metasploit correctly to work with the chosen settings. The launching and configuration of Metasploit are shown in Figure 15 and Figure 16.

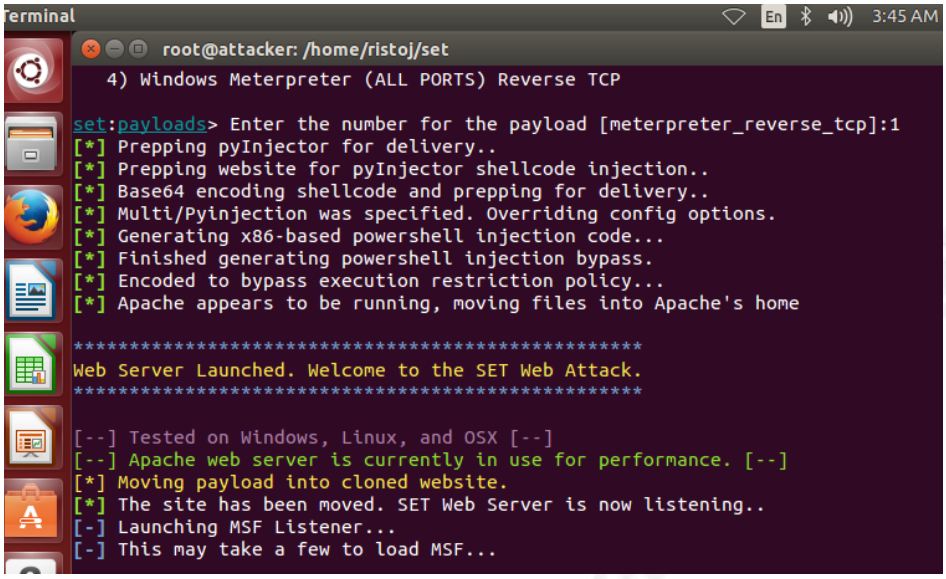

Figure 15: SET Java website completion and launching of applications

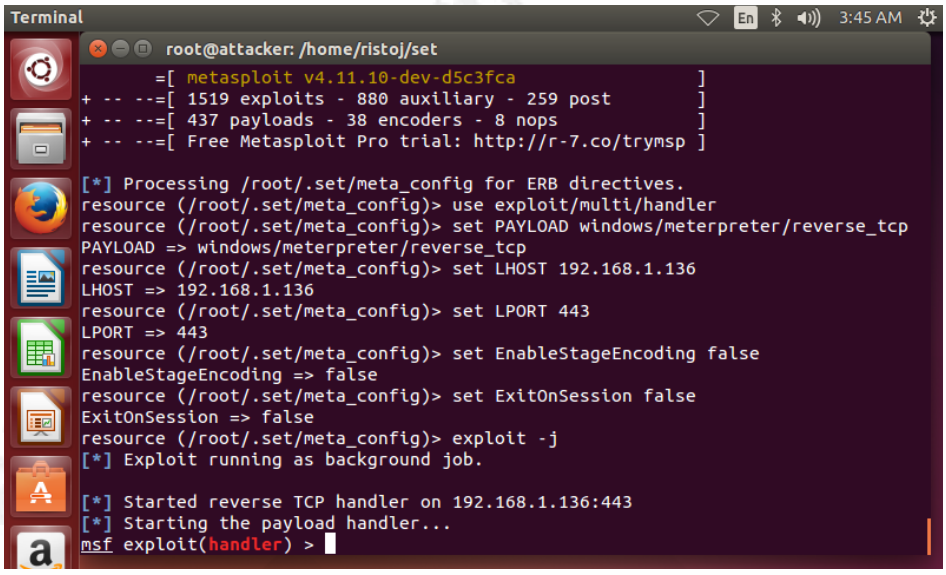

Figure 16: Metasploit setup for Java website attack

Once the site is launched, the website that SET created and loaded looks similar to the one shown in Figure 17.

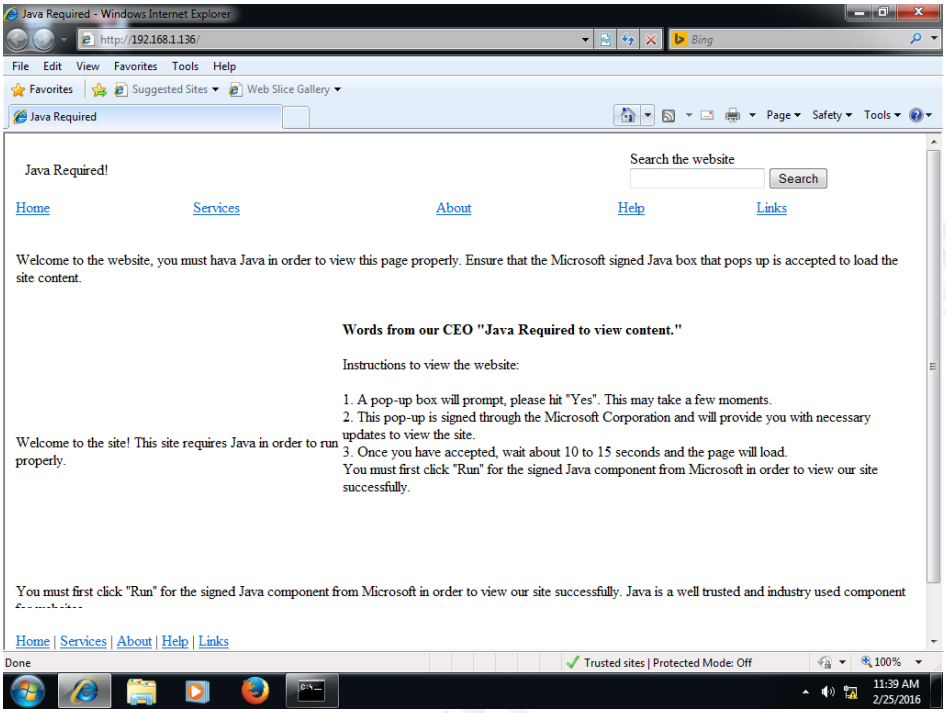

Figure 17: Target web browser view in Java attack

For the attack to work, some user interaction is needed. Just browsing to the website is not enough. The user, after approximately 15 seconds, is presented with a dialogue box similar to the one shown in Figure 18. The user will need to both choose the "I accept the risk" checkbox and then select the Run button to launch the attack.

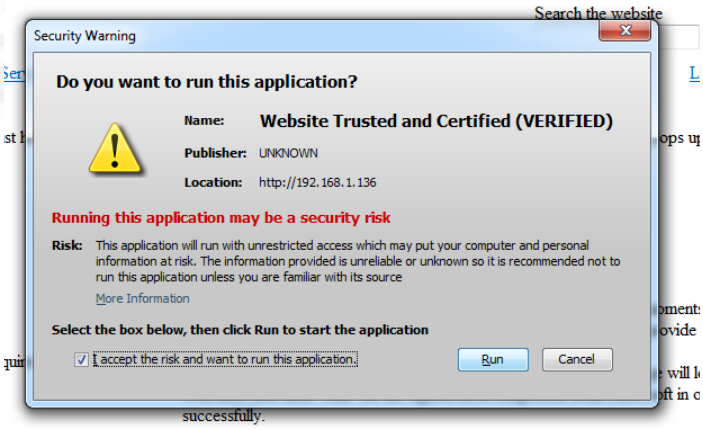

Figure 18: Java popup displayed to user in Java attack

Author Name, email@address

After the user selects "Run" from the dialogue box, the attack is in its final stages of completion. All that is needed is for the supplied code to inject properly into the end user system, and connect back to the attacker's listener. The results are displayed in Figure 19.

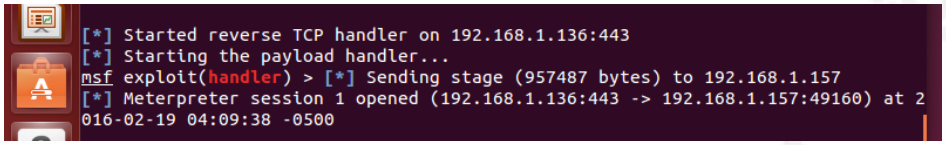

Figure 19: Success of the Java website attack

Confirmation of the system specific information is presented in Figure 20.

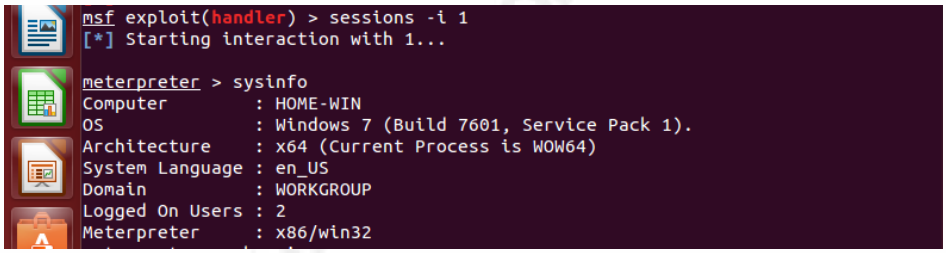

Figure 20: Website attack confirmation

For this attack, the multiple attacks were performed by having the target system browse to the site, accepting and running the application as presented to the end user.

3.3.3. Internet Explorer Vulnerability MS11-003

The third vulnerability used in the testing was an Internet Explorer vulnerability, specifically the MS11-003. This vulnerability was first reported publically published December 8th, 2010 and was patched on February 8th, 2011. Security vendors started providing signatures for detecting this in mid-December 2010, as exploits were being seen in the wild (IBM, 2010). This provided a window of over 60 days for attacks prior to a solution being provided by the vendor (Krebs, 2012). Within Metasploit, the exploit

that takes advantage of this vulnerability is found in "exploit/windows/browser/ms11_003_ie_css_import".

Several pieces of information need to be entered to configure this exploit. First, the IP address of the web server that will host the exploit must be identified. In the testing performed, it was again 192.168.1.136. This value was configured for the SRVHOST option. Other parameters can be left to the default values, and the exploit will function. However, within the testing done, the payload was also specifically named as the windows/meterpreter/reverse_tcp payload. The only other option configured was the lhost value, which was set to 192.168.1.136.

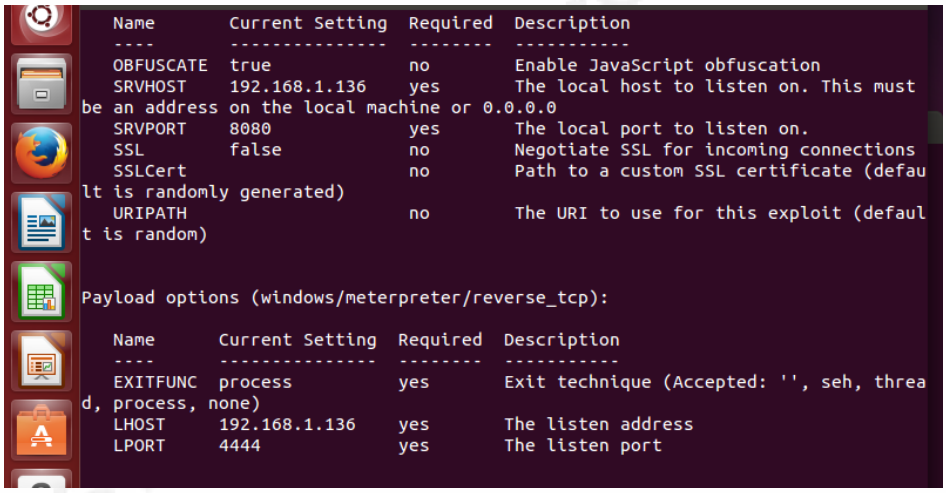

Figure 21: MS11-003 options

With these parameters, the exploit is launched by running the exploit command. Metasploit properly initiates the website using the Apache web server on the attacker computer and creates the listener within Metasploit for the successful connections. This is shown in Figure 22.

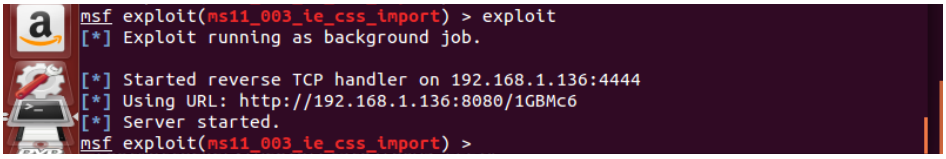

Figure 22: Successful Exploit launch with URL information

For this attack to function, the targets web browser must connect to this newly created website. In actual attacks, this could be achieved through phishing emails containing the link. Regardless of how the information is delivered to the end user, it was the success rates of this type of attack that were being measured, not how the attack was being distributed. In the testing, the URL shown in Figure 22 was typed into the target's web browser to initiate the web browser connection to the affected website. This connection was performed 20 times during the testing, with successful meterpreter sessions being recorded.

3.3.4. Adobe Acrobat Vulnerability CVE 2010-1240

The fourth type of exploit used in testing was related to the Adobe Acrobat software. This software was selected due to the ubiquitous nature of Portable Document Format (PDF) files, the free availability of the Adobe Reader product, and the availability of exploits within the selecting testing tools. The experimentation exploited the particular vulnerability discussed in CVE2010-1240. Fortinet reported that this vulnerability in June 2010 ranked second for overall malware activity in May 2010 (IT Brief, 2010).

Within Metasploit, this particular vulnerability is found by using the "exploit/windows/fileformat/adobe_pdf_embedded_exe" exploit. A second version is also available that doesn't use javascript, but this was not utilized in the testing.

This exploit requires minimal configuration. The payload selected for testing was "windows/meterpreter/reverse_tcp", with the lhost value set to 192.168.1.136. This is shown in Figure 23

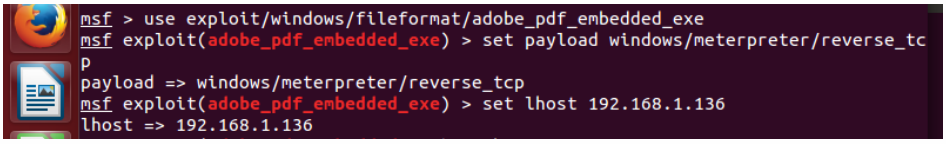

Figure 23: PDF exploit use

Otherwise, the remaining values were left at default. Upon running this exploit, by using the Metasploit exploit command, a file called evil.pdf is created and stored on the attacker's computer. This is shown in Figure 24. The pdf needs to move from the attacker system to the target system. In a normal attack, this could be accomplished

through a phishing campaign or by an infected website. For the testing performed, the file was moved to the target system by placing it onto a USB device, and loading the file directly onto the desktop of the target system.

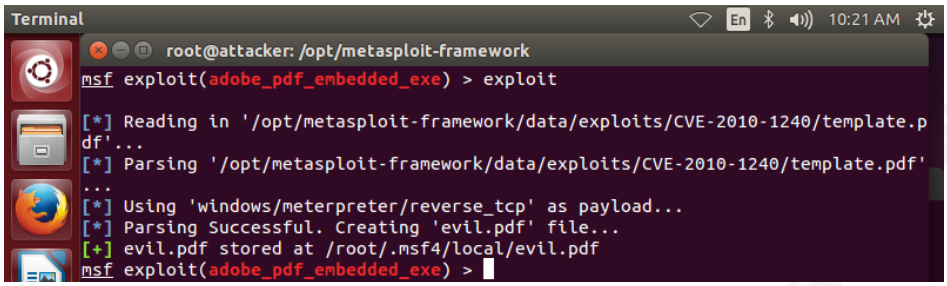

Figure 24: PDF file creation

This exploit also does not automatically create the listener on the attacker's computer. "Exploit/multi/handler" was used to provide the listener functionality, and the payload was set to "windows/meterpreter/reverse_tcp". The values for lhost and lport were set to 192.168.1.136 and 4444 respectively. This configuration process is displayed in Figure 25. With these values configured and the listener started, the settings needed for the attacker system were complete.

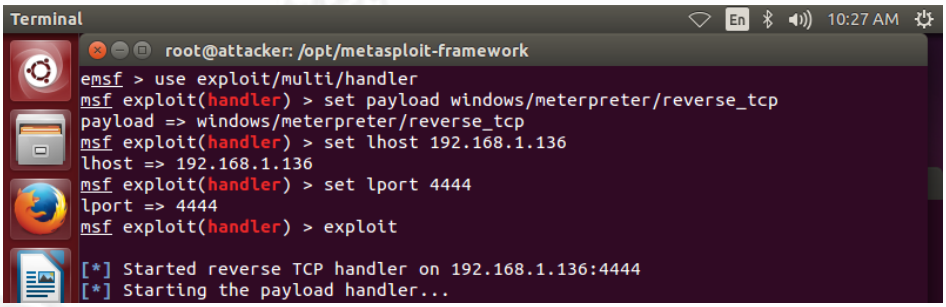

Figure 25: Configuring and starting the listener

With the infected PDF file on the target computer, all that needed to be done was to open the infected file. When this was performed, a dialogue box appears on the target system similar to that shown in Figure 26.

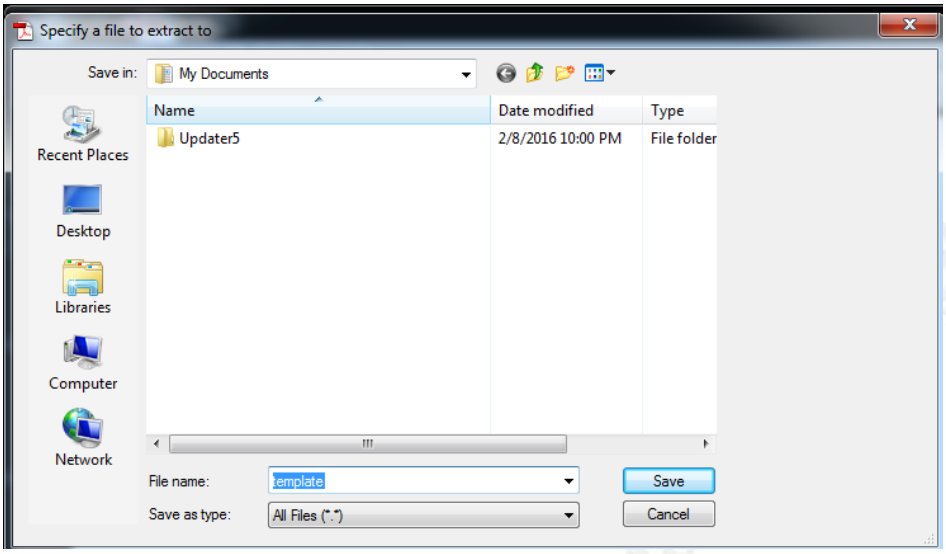

Figure 26: Infected PDF dialogue box

With the minimal information presented to the user, selecting save will continue the exploit. If that is done, a second dialogue box will appear, similar to that shown in Figure 27. The second paragraph of this message does provide information to the user that to view the information, the need to select the checkbox option of "Do not show this message again" and press Open. After this message, it also states that the file may contain macros, viruses, etc. that could potentially harm the computer. This warning is provided by the Acrobat program, to provide further options for the end user to not run a potentially malicious piece of software.

By selecting the Open button, the exploit is completed, and the connection is initiated back to the attacker computer. For the success rate testing, this file was opened 20 times on the target system to determine the success rate of this exploit.

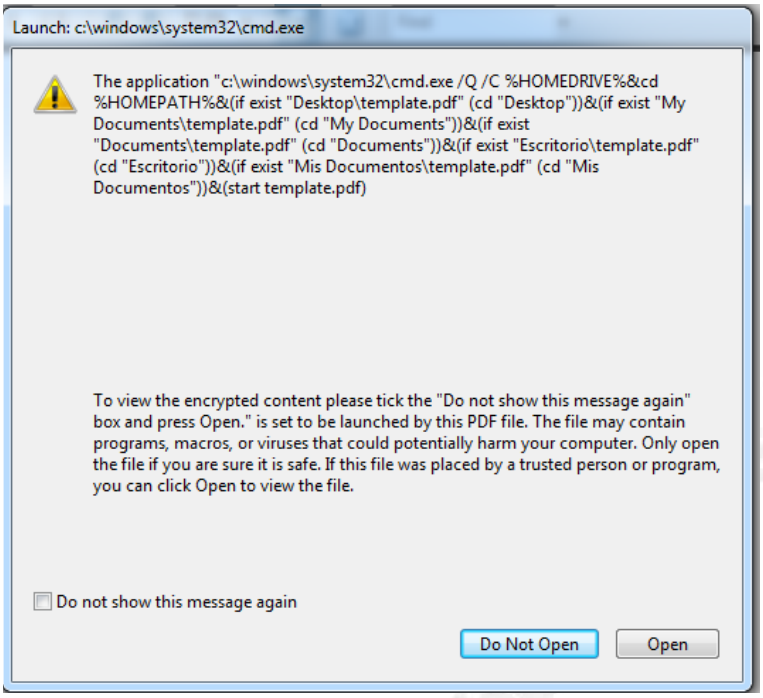

Figure 27: Infected PDF permission to open

3.3.5. Firefox Bootstrap Vulnerability

The fifth exploit tested as part of this work targets functionality within client programs. Firefox, a popular alternative web browser (NetMarketShare, 2016), provides the ability to add new functionality to the browser through something called extensions. In the exploit being used for the testing, Metasploit takes advantage of the ability to automatically load an extension from a website and have the user execute it. The add-on in question runs with the user's permissions but creates the backdoor to the attacker computer. This exploit is found within Metasploit in "exploit/multi/browser/firefox_xpi_bootstrapped_addon".

Configuration for the exploit only requires that the SRVHOST value is set correctly. The exploit automatically sets the "generic/shell_reverse_tcp" payload, and value of LHOST is set to localhost of the machine that Metasploit is running on. For the testing, the value of 192.168.1.136 was set both for SRVHOST and LHOST parameters. All other values were left at default. The configuration steps are shown in Figure 28. Upon execution of this exploit, a website is created using the Apache software on the

Attackers Computer, running on the SRVHOST address, and the default port is 8080 with a random file name. An example of the URL created is http://192.168.1.136:8080/91vhM6v. The random file name can be removed if desired through the configuration options within Metasploit.

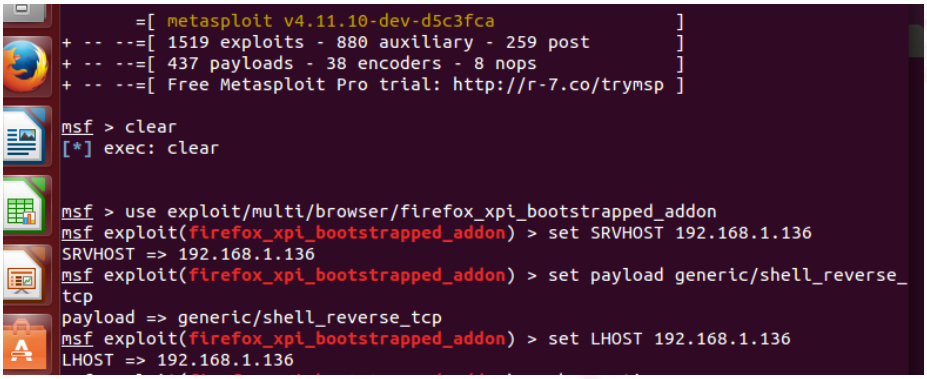

Figure 28: Firefox configuration steps

Similar to the PDF attack described previously, this exploit does not automatically create a listener on the attacker computer. For this testing, the exploit/multi/handler was used to permit connections to be established to the attacker computer.

With the website created by Metasploit above, all that is needed is to have a target system access the site in question. For the testing performed, this was done by manually entering the website address into Firefox. Once the browser connects to the site, the site displays that an add-on is needed to view the site, and presents an install link. In addition, Firefox displays a message that states it prevented the site from installing software on the computer, but also provides you with the option of allowing the install. Once the user installs the add-on, the connection back to the attacking computer is initiated, and access is gained. How this information is presented to the end user on the default websites is shown in Figure 29.

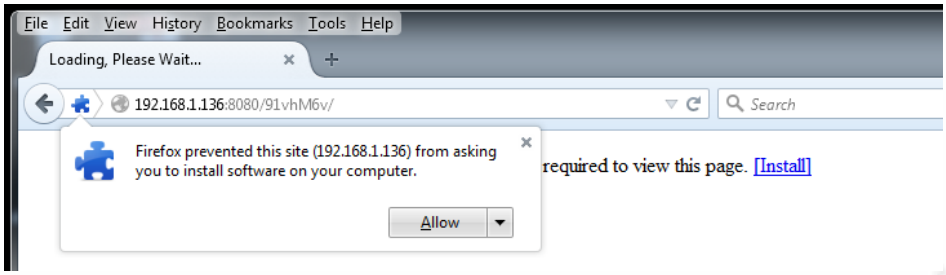

Figure 29: Firefox add-on dialogue boxes

For the tests performed, this website was accessed 20 times, and each time the add-on was installed on the target system. If a successful connection back to the attacker computer was created, the attack was marked as successful.

3.4. Results

Each exploit was run using the procedures described in sections 3.3.1 through 3.3.5, and the results were recorded as successes and failures. This information is captured in Table 1.

Exploit	Attempts	Successes	Success Rate %
Malicious Executable	20	19	95%
Java Download	20	20	100%
Internet Explorer (MS11_003)	20	19	95%
Adobe Acrobat (CVE2010-1240)	20	20	100%
Firefox bootstrap	20	20	100%

Table 1 - Testing results

4. Analysis

The results presented in Table 1 provide an interesting view into client-side exploits. While not an exhaustive testing suite, it indicates that client-side exploits

function practically 100% of the time. This is a surprisingly stark contrast to the results found in the work on server-side exploits by Dondo et al. in 2014.

Client-side exploits tend to have the target program execute a regular function but in an unexpected way that the original programmer did not envision. The functionality built into the system is abused by the attacker, but the user is involved and often granting permission. In contrast, server-side exploits often are breaking out of the program functionality to enable execution of the attacker's code. These are, by their nature, more problematic for successful running.

The results presented in Table 1, show that client-side exploits run successfully nearly all of the time. While the sample size for this testing is small, it does indicate that client-side exploits, if they are within the target network, will successfully run if permitted to execute.

5. Conclusion

Attacks continually bombard the networks that we defend today. They vary in type and variety, but a large portion of the attackers choose to target the end user in some way to gain their access to the organization. By targeting the user, usually an independent entity in the corporation, attackers attempt to bypass the signatures, notifications and blocking that defender deploy and maintain their networks. Focusing on the human factor, attackers can appeal to numerous emotions or anxieties of the user to have them perform an action, allowing the attacker into the network.

By looking at the client side attack possibilities, this work attempted to see if additional information could be used by defenders to help prioritize their work on these client-side vectors into the network. Based on the testing performed, it was shown that the success rates were between 95% and 100% for all items tested. While not an exhaustive sample size, further testing and analysis on this topic are needed to investigate if this result holds in the majority of client-side exploits. The results also reinforce the need for end user awareness and training to avoid end users being tricked into running these programs. The results show that if client-side attacks are sent successfully into the

target network, it is only the user, by choosing to run the file in question or not, that stands between the attacker's success or failure.

Based on the research performed, defenders should focus their efforts on the most damaging and relevant vulnerabilities within their environment for client-side attack vectors. The results show that the degree of success of client-side exploits is not based on the technology in use, but rather the ability of the attacker to persuade the end user to run the program in question in the first place. If the attacker is successful there, the malicious software will almost certainly work and provide them with access when executed on the target system.

Author Name, email@address

References

APWG. (2015, December 23). *Phishing Attack Trends Report - 1Q-Q3 2015.* Retrieved February 4, 2016, from APWG: http://docs.apwg.org/reports/apwg_trends_report_q1-q3_2015.pdf

Bhatt, S., Horne, W., & Rao, P. (2011). On Computing Enterprise IT Risk Metrics. *26th IFIP TC 11 International Information Security Conference* (pp. 271-280). Lucerne: Springer Berlin Heidelberg.

Deschatres, S. (2014, July 8). *Social Engineering: Attacking the Weakest Link in the Security Chain*. Retrieved February 4, 2016, from Symantec Official Blog: http://www.symantec.com/connect/blogs/social-engineering-attacking-weakest-link-security-chain

Dondo, M., Risto, J., & Sawilla, R. (2014). *Reliability of exploits and consequences for decision support.* Ottawa: DRDC. Retrieved from http://cradpdf.drdc-rddc.gc.ca/PDFS/unc194/p801970_A1b.pdf

IBM. (2010, December 16). *Microsoft Internet Explorer CSS Remote Code Execution*. Retrieved March 13, 2016, from IBM Threat Security Systems: http://www.iss.net/threats/404.html

IT Brief. (2010, June 07). *Fortinet's Latest Threat Report Shows New PDF Exploit*. Retrieved from IT Brief: https://itbrief.co.nz/story/fortinets-latest-threat-report-shows-new-pdf-exploit/

Kaspersky Lab. (2014). *Spam and Phishing Statistics Report Q1-2014*. Retrieved March 2, 2016, from Kaspersky Lab: https://usa.kaspersky.com/internet-security-center/threats/spam-statistics-report-q1-2014#.Vtjo528UWCh

Krebs, B. (2012, October 12). *In a Zero-Day World, It's Active Attacks that Matter*. Retrieved March 13, 2016, from Krebs On Security: http://krebsonsecurity.com/2012/10/in-a-zero-day-world-its-active-attacks-that-matter/#more-16949

Author Name, email@address

Net Applications. (2016, January). *Desktop Operating System Market Share*. Retrieved February 5, 2016, from Market Share Statistics for Internet Technologies: https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0

NetMarketShare. (2016, February 16). *Desktop Browser Market Share 2015*. Retrieved February 16, 2016, from NetMarketShare: https://www.netmarketshare.com/browser-market-share.aspx?qprid=0&qpcustomd=0&qpsp=2015&qpnp=1&qptimeframe=Y

Palo Alto Networks. (2014, December 10). *Threat Landscape Review.* Retrieved February 4, 2016, from Palo Alto Networks: https://www.paloaltonetworks.com/content/dam/paloaltonetworks-com/en_US/assets/pdf/reports/Unit_42/threat-trend-threat-landscape-review.pdf

Symantec. (2015, April). *Internet Security Threat Report, Volume 20.* Retrieved February 4, 2016, from Symantec: https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf

Verizon. (2015, April 26). *2015 Data Breach Investigations Report.* Retrieved February 4, 2016, from Verizon: http://www.verizonenterprise.com/DBIR/

Author Name, email@address