EXPLORING VULNERABILITIES IN NETWORKED TELEMETRY

Authors: Felix Shonubi, Ciara Lynton, Joshua Odumosu, Daryl Moten
Advisors: Dr. Richard Dean, Dr. Farzad Moazzami and Dr. Yacob Astatke
Department of Electrical and Computer Engineering
Morgan State University

ABSTRACT

The implementation of Integrated Network Enhanced Telemetry (iNET) in telemetry applications provides significant enhancements to telemetry operations. Unfortunately such networking brings the potential for devastating cyber-attacks and networked telemetry is also susceptible to these attacks. This paper demonstrates a worked example of a social engineering attack carried out on a test bed network, analyzing the attack process from launch to detection. For this demonstration, a penetration-testing tool is used to launch the attack. This attack will be monitored to detect its signature using a network monitoring tool, and this signature will then be used to create a rule which will trigger an alert in an Intrusion Detection System. This work highlights the importance of network security in telemetry applications and is critical to current and future telemetry networks as cyber threats are widespread and potentially devastating.

KEY WORDS

Telemetry, Exploit, Metasploit, Intrusion Detection System (IDS)

I. INTRODUCTION

Computers are used in telemetry communication as the brain behind wireless transmission and reception of data. They are used to process and store the received data, making them vital to telemetry [1]. The telemetry process is valued in being remote and mechanized; therefore, the true status of the equipment is not manually monitored. A security breach on a telemetry network is especially threatening as the data is collected at inaccessible points and transmitted to receiving stations [1].

Systems that require external instructions are susceptible to volatile security breaches. The attacks performed in our lab show how easy it is to infiltrate a network system that is not constantly monitored. Computers are an important part of network systems but are susceptible to cyber-attacks and this also puts networked telemetry at risk. As a result, there is a potential for cyber-attackers to compromise telemetry networks in a manner similar to the approach shown in this paper. With the changes we suggest in this paper, telemetry networks can work towards being more secure.

II. BACKGROUND

WINETS TEST BED OVERVIEW

Cybersecurity is one of three major research areas in Morgan State's Wireless, Networks and Security (WiNetS) laboratory. The cybersecurity research unit conducts all of its security-related experiments on an isolated local area network connection, also known as a test bed. This is done so that cyber-attacks are not accidentally carried out on a live network. The test bed is equipped with the necessary tools to explore various cyber security disciplines such as penetration testing, vulnerability assessment, intrusion detection and prevention, and network analysis.

All the computers on the test bed are connected to a central OSI layer-2 switch. This switch is configured to enable the computers to communicate efficiently with each other on the network. Since the network is isolated, static IP addresses are assigned to the hosts. The test bed has about ten computers set up with various software for attacking, detecting, monitoring, and visualizing several activities on the network. The main tool used on attack machines is Metasploit, which is a penetration testing tool. An intrusion detection system is set up on one of the computers to detect attacks on the network. Also, a mail server is configured to allow cyber-attacks such as phishing emails and social engineering attacks to be carried out on the test bed. In addition, a network packet analyzer is set up to capture network packets and data traffic for analysis purposes and a network visualization system is also employed on the test bed. The WiNetS test bed is a scalable setup that allows more computers and software to be added as needed. A schematic of the test bed in the WiNetS lab is shown in Figure 1.

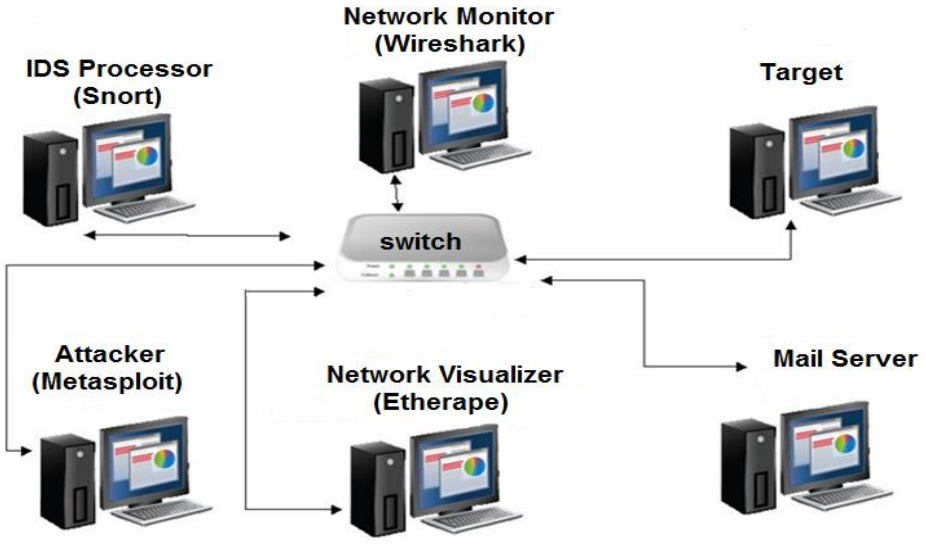


Figure 1: WiNetS Lab Test Bed Network

Three main software products are vital to the work demonstrated in this paper: Metasploit, Wireshark, and Snort. Metasploit resides on the attack computer and is a well-known open source penetration testing tool. It is a database of many exploits and payloads and also has

information gathering and vulnerability assessment capabilities. It is used as the main attack tool in this paper. Wireshark is a network analyzer that is used for monitoring and troubleshooting networks. It is also open source and very popular in many industries. It captures data packets in real time and displays them in a form that makes analysis convenient. It is used to capture and analyze the cyber-attack used in this experiment. Finally, Snort is an open source intrusion detection tool which processes data on a network and matches it against known attacks. It is configured as network-based IDS where it is able to monitor a network and detect intrusions on it.

VULNERABILITY AND EXPLOIT

The vulnerability being exploited in this experiment is the MS10-046 which, according to the Common Vulnerabilities and Exposures website, allows local users or remote attackers to execute arbitrary code via a crafted .LNK or .PIF shortcut file which is not properly handled during icon display in Windows Explorer [2]. This vulnerability arises as a result of the mishandling of .LNK files when shortcut icons are displayed. The specific vulnerability lies in the Windows Shell component Shell32.dll which is a library responsible for parsing .LNK files and their icons. The Shell32.dll uses the LoadLibraryW function which reads the icon library file with execute rights and calls a function in the icon library, thus it executes possibly malicious code when trying to fetch icon data [3]. Although security updates for this vulnerability exist, unpatched versions of Windows XP Service Pack 3, Windows Vista, Windows 7, and some versions of Windows server are susceptible to this attack. According to Microsoft Security TechCenter, an attacker who successfully exploits this vulnerability could gain the same user rights as the local user [4]. This vulnerability is critical and severe in nature and is perfect for this experiment.

The exploit used in this experiment is the “Microsoft Windows Shell LNK Code Execution” which is a browser-type Metasploit exploit module. It has an “Excellent” rank in Metasploit for its reliability in exploitation. The exploit creates a Web Distributed Authoring and Versioning (WebDAV) service which allows users to perform remote editing of websites over HTTP or HTTPS connections [5]. This service can be used to run a payload when accessed as a UNC path [6]. Succinctly, this exploit starts a server and generates a UNC path link pointing to the malicious code on the attack computer. The attacker only needs a target to open this link on a vulnerable computer and the exploit will occur.

III. PROCEDURE

ATTACK METHODOLOGY

The exploit is first configured in Metasploit by selecting the exploit to be used, in this case “MS10-046”. Next, the payload to be delivered is selected. A payload is simply the code that is sent to the target computer after exploitation. The payload used in this experiment is a reverse connection Meterpreter payload. Meterpreter is a robust feature of Metasploit which gives an attacker the ability to control the target computer via terminal-like commands. Meterpreter operates via dll injection and thus resides completely in the memory of the remote host (target)

and leaves no traces on its hard drive, making it difficult to detect [1]. When the exploit is run, a link which contains the attacker's IP address attached to some random text is generated. This link points to the WebDAV service hosted on the attacker's computer. Because an IP address can clearly be seen in the link, it is very unlikely that a target user would click on the link. An easy solution is to disguise this link in a hyperlink pointing to a popular or harmless website or just plain text as shown in Figure 2 below. In the example below, the link generated by Metasploit is disguised as "CLICK HERE" which is less ominous than the initial link.

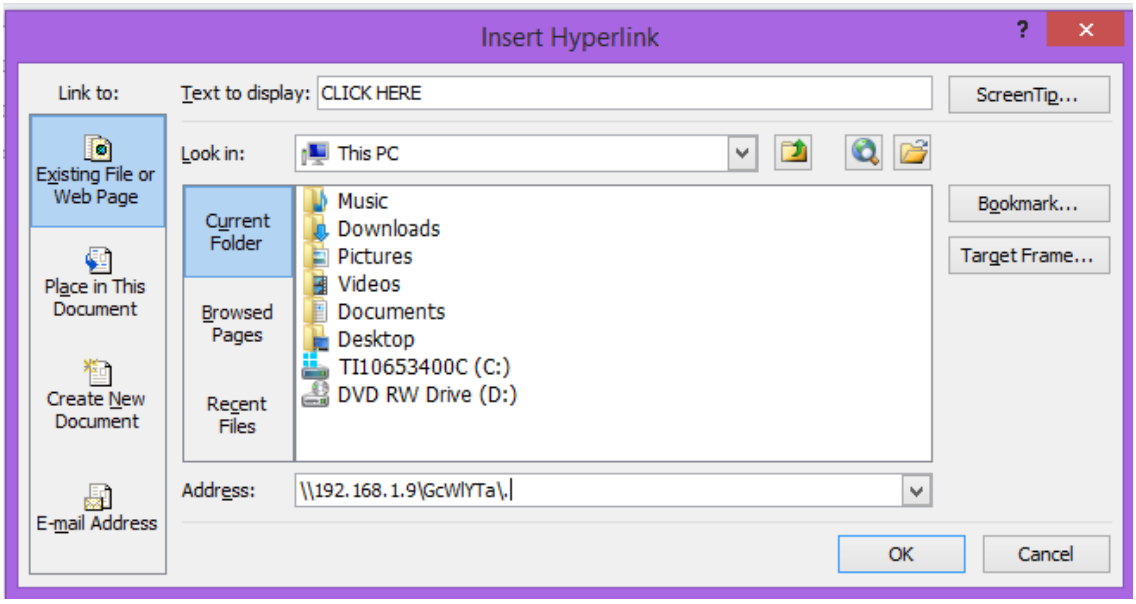


Figure 2: Disguising a Malicious Link Using a Hyperlink

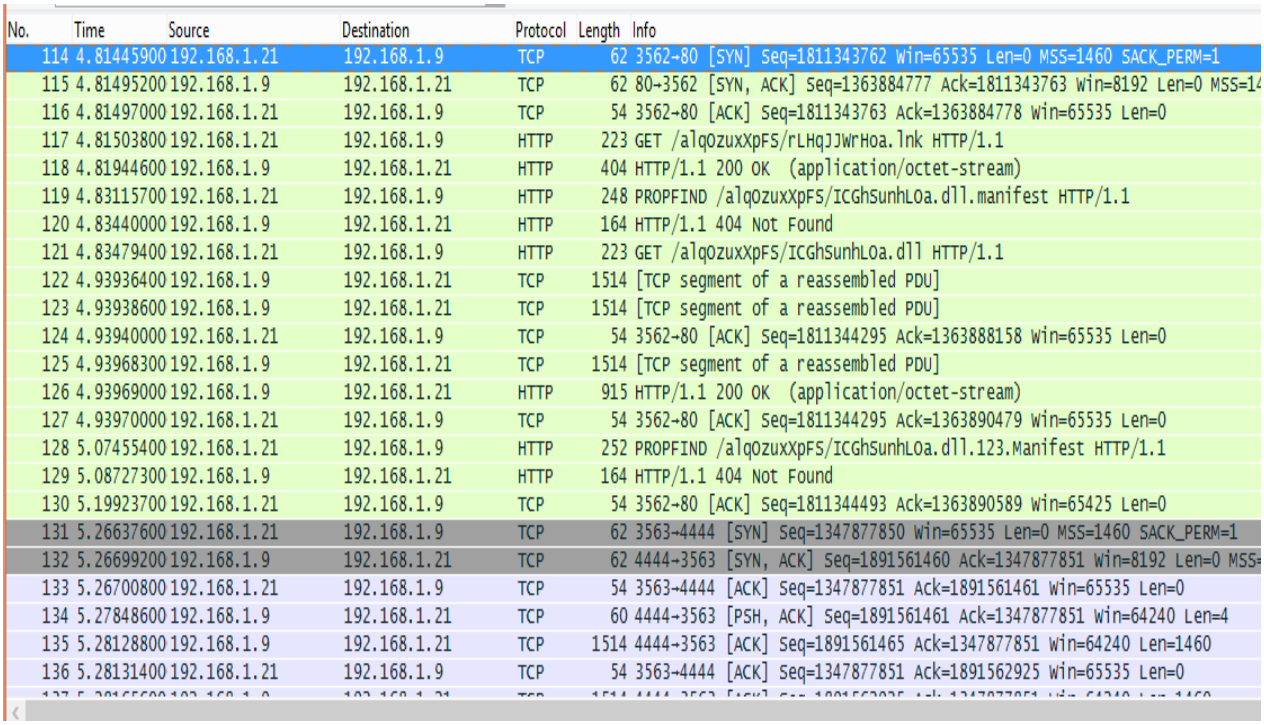
This text is included in an email, requiring the target to click on the link and this scenario is the basis of most common social engineering attacks. This type of attack depends on user interaction and can trick unsuspecting victims. If a user on the target computer opens the link, a connection request is made to the attack computer and the attack computer begins to send malicious code almost instantaneously back to the target. After this code is sent, a Meterpreter session is opened on the attack computer if the exploit is successful. Meterpreter allows the attacker to run file system commands which allow viewing, editing, deleting, uploading, and downloading of files on the target computer. The attacker can also execute commands remotely and make various system changes to the target computer. The attacker can even cover his/her tracks by clearing system event logs.

In a larger scale attack such as an attack on a telemetry network, the attackers' main focus would be to escalate privileges or to pivot to other hosts on the network with more administrative access. It is much harder to get to highly protected servers or hosts than it is to hack into vulnerable users on the same network. If an attacker can successfully infiltrate a vulnerable computer on the network, the whole network is compromised, even if the infiltrated computer belongs to a low-level user. A hacker can pivot from this system to other systems with higher privileges on the network or on connected networks. Escalating privileges gives the hacker the

ability to perform actions on the system with fewer restrictions and to run applications and perform operations that have a bigger impact on the system [7].

PACKET LEVEL ANALYSIS OF ATTACK

This section of the paper focuses on the analysis of this cyber-attack by examining the data generated by the attack at the packet level. The network monitoring tool, Wireshark, is used to monitor and capture the data from the attack. The data stream begins with a TCP three-way handshake between the target and the attack computer, initiated by the target computer. This handshake occurs as soon as the target user clicks on the emailed link generated by Metasploit. The following packets show the target user requesting to connect to the shared path which is the address on the link hosted on the attack computer. After this request is authenticated, the target computer then sends another request to access the content located at the address which it supposes to be a shortcut icon but is in fact exploit code. This is shown in packets 117 and 121 in Figure 3 below. The attack computer then sends the exploit code (packets 122,123, and 125) and finally the payload (Meterpreter) to the target computer (packets 135 and beyond). At this point, the attacker has complete control over the target computer.



No.	Time	Source	Destination	Protocol	Length	Info
114	4.81445900	192.168.1.21	192.168.1.9	TCP	62	3562→80 [SYN] Seq=1811343762 Win=65535 Len=0 MSS=1460 SACK_PERM=1
115	4.81495200	192.168.1.9	192.168.1.21	TCP	62	80→3562 [SYN, ACK] Seq=1363884777 Ack=1811343763 Win=8192 Len=0 MSS=14
116	4.81497000	192.168.1.21	192.168.1.9	TCP	54	3562→80 [ACK] Seq=1811343763 Ack=1363884778 Win=65535 Len=0
117	4.81503800	192.168.1.21	192.168.1.9	HTTP	223	GET /alqOzuxXpFS/rLHqJJWrHoa.lnk HTTP/1.1
118	4.81944600	192.168.1.9	192.168.1.21	HTTP	404	HTTP/1.1 200 OK (application/octet-stream)
119	4.83115700	192.168.1.21	192.168.1.9	HTTP	248	PROPFIND /alqOzuxXpFS/ICGhSunhLOa.dll.manifest HTTP/1.1
120	4.83440000	192.168.1.9	192.168.1.21	HTTP	164	HTTP/1.1 404 Not Found
121	4.83479400	192.168.1.21	192.168.1.9	HTTP	223	GET /alqOzuxXpFS/ICGhSunhLOa.dll HTTP/1.1
122	4.93936400	192.168.1.9	192.168.1.21	TCP	1514	[TCP segment of a reassembled PDU]
123	4.93938600	192.168.1.9	192.168.1.21	TCP	1514	[TCP segment of a reassembled PDU]
124	4.93940000	192.168.1.21	192.168.1.9	TCP	54	3562→80 [ACK] Seq=1811344295 Ack=1363888158 Win=65535 Len=0
125	4.93968300	192.168.1.9	192.168.1.21	TCP	1514	[TCP segment of a reassembled PDU]
126	4.93969000	192.168.1.9	192.168.1.21	HTTP	915	HTTP/1.1 200 OK (application/octet-stream)
127	4.93970000	192.168.1.21	192.168.1.9	TCP	54	3562→80 [ACK] Seq=1811344295 Ack=1363890479 Win=65535 Len=0
128	5.07455400	192.168.1.21	192.168.1.9	HTTP	252	PROPFIND /alqOzuxXpFS/ICGhSunhLOa.dll.123.Manifest HTTP/1.1
129	5.08727300	192.168.1.9	192.168.1.21	HTTP	164	HTTP/1.1 404 Not Found
130	5.19923700	192.168.1.21	192.168.1.9	TCP	54	3562→80 [ACK] Seq=1811344493 Ack=1363890589 Win=65425 Len=0
131	5.26637600	192.168.1.21	192.168.1.9	TCP	62	3563→4444 [SYN] Seq=1347877850 Win=65535 Len=0 MSS=1460 SACK_PERM=1
132	5.26699200	192.168.1.9	192.168.1.21	TCP	62	4444→3563 [SYN, ACK] Seq=1891561460 Ack=1347877851 Win=8192 Len=0 MSS=
133	5.26700800	192.168.1.21	192.168.1.9	TCP	54	3563→4444 [ACK] Seq=1347877851 Ack=1891561461 Win=65535 Len=0
134	5.27848600	192.168.1.9	192.168.1.21	TCP	60	4444→3563 [PSH, ACK] Seq=1891561461 Ack=1347877851 Win=64240 Len=4
135	5.28128800	192.168.1.9	192.168.1.21	TCP	1514	4444→3563 [ACK] Seq=1891561465 Ack=1347877851 Win=64240 Len=1460
136	5.28131400	192.168.1.21	192.168.1.9	TCP	54	3563→4444 [ACK] Seq=1347877851 Ack=1891562925 Win=65535 Len=0

Figure 3: Wireshark Capture Showing Attack Details

It should be noted that it takes about one second from the time the link is clicked to the time the exploit is completed and another four seconds for the Meterpreter shell to be opened on the attack computer. This portion of the experiment is necessary because the signature of the attack can be obtained from this analysis as will be shown in the next section of this paper,

DETECTION OF ATTACK

The analysis of the attack as shown above yields important information that can be used to detect the attack. This is done using Snort, the Intrusion Detection System (IDS) used in this experiment. To detect the attack, a signature-based rule must be written in the Snort rules file, and this rule would have to match a portion of the data contained in the attack. The rule has to be as specific as possible to eliminate false positives and still detect this attack. On examining the attack data, specifically the portion of the data containing the exploit, certain keywords can be extracted and used to write a rule. The keywords extracted are "lnk" and "dav" which are shown in the data stream in Figure 4 below

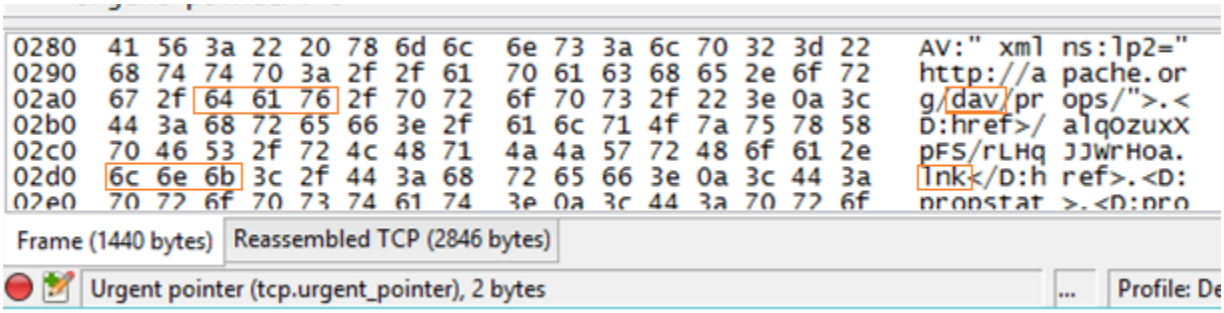


Figure 4: Highlighted Keywords Used in Creating IDS Rule

The "lnk" keyword represents the vulnerability being exploited in this attack. Even if the attacker changes the exploit code, the vulnerability being exploited is still present so it makes sense to trigger an alert based on the vulnerability. On the other hand, the "dav" keyword represents the WebDAV service mentioned above which hosts the exploit and payload on the attack machine. This server is a core component of this Metasploit attack and is accessed when the target user clicks on the emailed link. Having both of these words in the same packet is very unlikely to occur during normal internet traffic, therefore allowing Snort to match these two words is a logical way to detect the attack. It should be noted that any part of the data from the attack, malicious or not, can be written to a rule and would also trigger an alert. However, if the rule is not properly written, it can result in a lot of false positives or even worse, a false negative. The rule written to detect this attack is shown in Figure 5. The two words used to match the exploit data are written in hexadecimal form, which is obtained from Wireshark and is highlighted in Figure 5 also. This rule successfully detected the cyber-attack and this is shown at the bottom of Figure 6.

```
alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"Microsoft
Windows Shell LNK Code Execution"; content:"|64 61 76|";
content:"|6c 6e 6b|"; sid:1020009;)
```

Figure 5: Snort Rue Used to Detect Attack

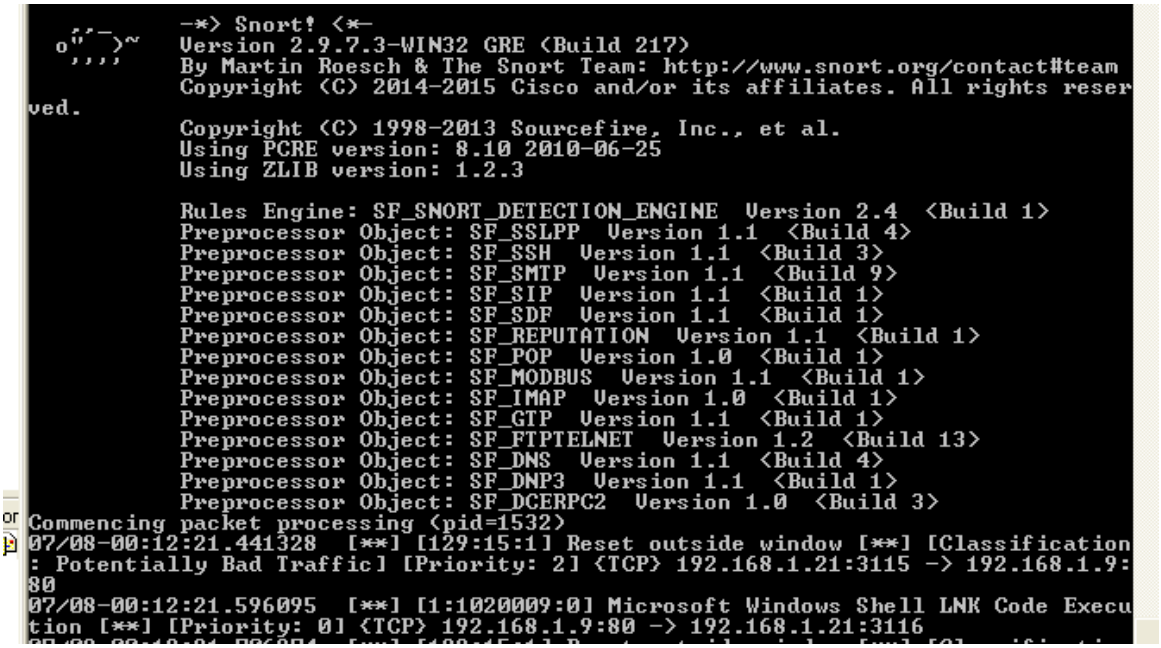


Figure 6: Snapshot of Snort in Operation Showing Detected Attack

This method of detecting cyber-attacks involves prior knowledge of the signature of the attack and is very common in the cybersecurity industry. Another method of detecting intrusions is by using an anomaly-based intrusion detection system. According to Bhuyan et al. [8], anomaly-based intrusion detection in networks refers to the problem of finding exceptional patterns in network traffic that do not conform to the expected normal behavior. This system works by determining what "normal" network behavior is, then comparing live traffic to this baseline to check for deviations. Both methods of intrusion detection have their own advantages over the other, but generally, most IDS use either signature-based, anomaly-based, or a combination of both systems.

IV. CONCLUSION

Networked telemetry undoubtedly provides improvements to current telemetry operations, but it also opens the door to possibly a major drawback of all computer networks. Cyber-attacks are prevalent in today's society and there is increasing consciousness across all industries to combat this menace. This paper demonstrated by example the vulnerability of future networked telemetry to cyber-attacks. Computers are an inevitable part of network systems and as a result an entire telemetry network system can be potentially compromised by a cyber-attack. This provides motivation for development by the telemetry community for cyber security architectures [9] and for modeling and planning for risk management [10] to mitigate the effects.

V. ACKNOWLEDGEMENTS

We would like to thank our advisors, Dr. Richard Dean, Dr. Farzad Moazzami, and Dr. Yacob Astatke for their support leading to this paper.

VI. REFERENCES

[1] http://www.cse.scu.edu/~cmehta/handouts/scubrief.pdf

[2] "About the Metasploit Meterpreter." Offensive Security. N.p., n.d. Web. 26 June 2015. <https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/.>.

[3] "How LNK Exploit Protection Works." N.p., 31 July 2010. Web. 26 June 2015. <https://hitmanpro.wordpress.com/2010/07/31/how-lnk-exploit-protection-works/.>.

[4] "Microsoft Security Bulletin MS10-046 - Critical." Security TechCenter. N.p., 18 Apr. 2014. Web. 26 June 2015. <https://technet.microsoft.com/library/security/ms10-046>.

[5] "WebDAV." The Official Microsoft IIS Site. N.p., n.d. Web. 26 June 2015. <http://www.iis.net/configreference/system.webserver/webdav>.

[6] "Microsoft Windows Shell LNK Code Execution." CVE-2010-2568. N.p., n.d. Web. 10 July 2015. <http://www.rapid7.com/db/modules/exploit/windows/browser/ms10_046_shortcut_icon_dllloader>.

[7] Oriyano, Sean-Philip, and Michael Gregg. Hacker Techniques, Tools, and Incident Handling, Jones & Bartlett Learning, Sudbury, MA, 2011, p179.

[8] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communication Surveys Tutorials*, vol. 16, no. 1, pp. 303, 2014.

[9] Moten, D, et al. Security Architecture for Telemetry Networks. ITC 2014, San Diego, CA

[10] Bazyar Shourabi, N et al. Model for Cyber Attack Risks in Telemetry Networks. ITC 2015, Las Vegas, NV