Snort IDS Ability to Detect Nmap and Metasploit Framework Evasion Techniques

Z. Jammes and M. Papadaki

Centre for Security, Communications and Network Research
Plymouth University, United Kingdom
e-mail: info@cscan.org

Abstract

Detecting exploit and port scan disguised by evasion technique is a challenge for IDS. This research examines the evasion technique provided by Nmap, a port scanner and Metasploit Framework, an exploit launcher against a famous IDS named Snort. The result tends to prove that Snort has the ability to detect port scan and exploit on condition to have a good configuration of Snort and signature for the exploit.

Keywords

IDS, Snort, Nmap, Metasploit Framework, evasion techniques, resilience

1 Introduction

Nowadays, information systems are increasingly open Internet. This opening is beneficial but is poses nevertheless a major problem: it brings a number of new attacks and requirement. The first effect is the implementation of a security policy around these systems. In addition to the implementation of firewalls and authentication systems are also necessary. To complete this security policy, it is also important to have monitoring tools to detect possible intrusions in the system. The solution is intrusion detection system but like each software, the IDS have also some weakness named: evasion techniques. Hopefully, over the time, the IDS are improved bringing new functionalities but therefore, they are become powerful but also difficult to configure. Today, the slightest error in configuration can then let go of thousands of intrusion without being alerted.

2 Evasion techniques

The evasion techniques were firstly introduced by Ptacek and Newham (1998). They explained that they described three evasions which are the foundations: the insertion, the evasion and the denial of service.

The insertion attack is an attack where IDS does not detect anything although on the target system, the attack does occur and the target system ignored the packets. The evasion attack is an attack where the target system accepts the packets although the IDS refused the packets. The aim of these evasion techniques are the packet content in the traffic was differently interpreted between the IDS and the end system; this

being due to the different system implementation. Finally, the denial of service attack is an attack is an attack with the aim of makes unavailable the IDS. This known evasions techniques target specific layers of the TCP/IP protocol stack and use their weakness (for instance fragmentation). Nowadays, these techniques have also spread to other different protocol as SMB, DCERPC and HTTP.

In 2010, Stonesoft (Boltz, Jalava, & Walsh, 2010) shared findings on a new evasion threat. Indeed, they discover this year new techniques to evade IDS named Advanced Evasions Techniques (AET). The AETs target multiple layers of the protocol TCP/IP stack and combine multiple evasion methods. Furthermore, they can be changed or modified during the exploit. The problem is that they do not conform to the rules used by IDS today.

Nowadays, many tools used to test the security implements different technical evasion. For instance, Nmap (2012) is designed to detect open ports, identify hosted services and information about the operating system of a remote computer but provided some evasion techniques. Metasploit Framework (Maynor , 2007) is a tool that allows launching different exploit against a remote host while also providing different evasion techniques. An exploit is a computer program to “exploit” a security flaw or vulnerabilities.

Snort (2012) is a signature-based IDS e.g. it uses signatures of known attack to detect the attack in the network traffic. It is very dependent signatures and therefore required to be updated regularly. Snort is also considered like anomaly-based IDS. It is able to detect some anomalies in the different protocol.

Snort is therefore based on the preprocessors to normalize traffic and detecting anomalies and on the rules to detect in this study exploits. preprocessors and rules will be put to the test.

3 Snort configuration against Nmap’s evasion techniques

The experiences made with Nmap can be easily redo because it does not necessary have specific equipment. The only requirement is to have 2 computer or virtual machine. The most important is to have one host which launches the scan and another which is scanned. It could be useful to prefer to target a Linux distribution rather than a windows system.

Nmap offers different scan techniques based on the TCP and UDP protocol. The sfPortscan is the preprocessor that is able to detect different port scan in function of its configuration. Most of the evasions are based on changes to the UDP, TCP and IP protocol. For this part, the experience uses different scans provide by Nmap.

The most efficient evasion technique provided by Nmap to evade this module is the fragmentation. Usually, fragmentation occurs when datagrams are larger than the allowable size, this limitation is called MTU (Maximum Transmission Unit). Each fragmented packet has an IP header for linking fragments together during the reconstruction.

Type of scan	With Frag3	Without Frag3
Syn scan/regular scan	OK	NO
Fin scan	NO	NO
Null scan	NO	NO
Maimon scan	NO	NO
Xmas scan	NO	NO
Connect scan	OK	NO
Ack scan	NO	NO
IP protocol scan	NO	NO
Intensive scan	OK	NO
Intensive scan plus UDP	OK	NO
Intensive scan all tcp	OK	NO
Slow comprehensive scan	OK	NO

Table 1 - Port scan detection with fragmentation

In this case, despite that the sfPortscan is enabled, Snort is unable to detect any port scan. Snort needs the frag3 preprocessor which performs the defragmentation of IP packets in order to prevent attack packets intentionally fragmented can escape detection. Snort is not able to detect some scans provided by Nmap. Indeed, the Fin, Null, Xmas, Maimon scan are not detected because this type of traffic does not exist normally on a network. In this case, it is important to add some rules to Snort such as:

```
alert tcp any any -> $HOME_NET any (msg:"FIN Scan";
flags: F; seq:1;)
alert tcp any any -> $HOME_NET any (msg:"NULL Scan";
flags: 0;)
```

The stream5 preprocessor is also an important piece to detect Nmap scan technique detection. It reconstructs TCP flows and it is also capable of reconstructing the UDP sessions. It allows rules to be executed on the data stream. Without it, once again, Snort cannot detect port scan.

SYN Scan	Detected
T5	OK
T4	OK
T3	OK
T2	OK
T1	OK
T0	NO

Table 2 - SYN scan detection with different timing

Another evasion technique, it is the possibility to choose the timing between sending two probes. Nmap provide different default template. The template names are paranoid (0), sneaky (1), polite (2), normal (3), aggressive (4), and insane (5). The first two are for IDS evasion. The paranoid mode waits 5 minutes between sending

each probe where the sneaky mode waits 15secondes. Without any difficulty, snort was able to detect the sneaky, polite, normal, aggressive and insane mode but it is not able for the paranoid mode. The problem is that sfPortscan analyse the packet on a windows of 60 second when the low sense level is selected. The best chance is to use the High level because it continuously track active host but it requires adjustments.

Nmap gives the possibility to change the TTL value in the different packet created. One thing to notice is if the TTL is set to 0, Snort is not able to detect any scan because it ignores each packet that has a TTL of 0. This evasion can be difficult to put in place in a real network because with a TTL of 0, it is impossible that a scan reaches their target, the different will be dropped before it happens. The only possible is to scan a computer in the same network so the attacker is inside the company or via a disgruntled employee.

Stream5, Frag3 and sFportscan are complementary and the best way to detect port scan with or without evasion techniques. After their effectiveness in detecting scans depend on their configuration.

4 Snort configuration against Metasploit's evasion techniques

In this part, it is important to configure this option like this: `config checksum mode: none`. Otherwise, the entire exploits tested are not detected because Snort seems to assume that the traffic with the bad checksum has no effect on the target.

For Metasploit Framework, the majority of the experiences needs different version of windows, old software version, and some specific software configuration. It is really difficult or impossible to retrieve older versions of software that some exploits target (Luckily it is impossible to find, in this way the average user is protected). Hopefully, the majority of exploits that target the browser (Internet Explorer) or some versions of OS are easily to recreate.

The evasion techniques used by the Metasploit Framework are evasions that are focus on HTTP, DCERPC, SMB, TCP protocol and HTML. In this part, Snort relies more on the rules than the preprocessor. The preprocessor are here to normalizes the traffic and make information that transits understandable and decipherable to ensure and increase the chances of detection.

Snort has static signatures so the different evasion techniques try to transform the exploit for that it stay understandable for the target but not for Snort e.g. the exploit does not match the signature of the rule.

Evasion	Netapi Exploit detection	Wkssvc Exploit detection
Without evasion	OK	OK
DCERPC::max_frag_size	OK	OK
SMB::pipe_evasion	OK	OK
SMB::pad_file_level	OK	OK
SMB::pad_data_level	OK	OK
TCP::max_send_size	OK	OK
TCP::send_delay	OK	OK

Table 3 - Exploit DCERPC/SMB detection

OK: Snort detects the exploit
NO: Snort does not detect the exploit

A first part of the evasion techniques take the advantage on some specificity of DCERPC, SMB and TCP protocols.

On the DCERPC protocol, it is possible to force the fragmentation of packet. In this condition, Snort is unable to understand the DCERPC protocol. Hopefully, the DCERPC2 preprocessor is able to defragment.

On the TCP protocol, it is possible to limit the size of the TCP segment. In this case, the packet are segmented and to be able to still see the exploit, Snort needs to have the stream 5 preprocessor activated.

A second part of the evasion techniques take the advantage on some specificity of HTTP protocols. This evasion allows changing some value in the HTTP header and encoding the HTTP body.

On the HTTP protocol, it is possible to compress the HTTP page to gain in bandwidth. The problem is that the IDS will not be able to detect the signature include in the HTTP body compressed. It is important to activate the inspect_gzip option. This options specifies the HTTP inspect module to “uncompress” the compressed data (in gzip/deflate) in HTTP response. With this option, Snort is able to still detect the exploit.

Metasploit framework is able to encode the HTTP body with different language such as base64,Unicode and JavaScript, Snort is not able to detect the exploit anymore because the rule does not recognize this type of encoding but Snort provides an alert if it detects the use of Base64 and JavaScript in the payload: “POLICY-OTHER base64-encoded uri data object found” and “Obfuscation JavaScript”. Normally, Snort is able to normalise the Unicode like the JavaScript but it could be difficult for Snort to refund the original code.

Evasion	IE Exploit detection	Mozilla Exploit detection
Without evasion	OK	OK
HTML ::base64	NO	Rejected
HTML::JavaScript::escape	NO	Rejected
HTTP::chunked	NO	NO
HTTP::compression	OK	Gzip:OK Deflate: Rejected
HTTP:junk_header	OK	OK
HTTP::server_name	OK	OK
SSL implementation	NO	NO

Table 4 - Exploit HTTP header and body detection

OK: Snort detects the exploit
NO: Snort does not detect the exploit
Rejected: the result obtained by the evasion technique is not sufficient

One of the weak points of Snort is its inability to detect an exploit in a traffic encrypted.

A third part the evasion techniques is more focus on the modification of URI, these evasions incorporate some evasion available with Nikto. Snort is able to send some alerts based on the anomalies found in the URI.

The problem is that Metasploit Framework according to the exploit offers different evasion options but it is important to highlights that in some case, when an evasion option is activated, it seems that the evasion option is not implemented and so did not make the modifications expected in the main packet of the exploit. Sometimes, the modifications worked but the exploit did not work anymore. The modification made prevents the exploit to work correctly due to changes too important. Certainly, it is possible that the exploit or the vulnerable software is not suitable for certain evasions but Metasploit Framework could at least warn users instead of proposing default evasion.

5 Conclusion and future work

Snort has the ability to detect most of the port scan made by Nmap and the exploit launched by the Metasploit Framework. For Nmap, Snort relies heavily on these preprocessors (Frag3, Stream5 and sfPortscan).

It is important to note with the default configuration provided on the official website of Snort that by default the sfPortscan is not activated. In this case, Snort is unable to generate an alert about port scan activities. An inexperienced user may believe to be protected but, in this case, Snort will not be able to generate alerts concerning scans Otherwise, the configuration provided for Stream5 and frag3 is sufficient to protect

and detect port scan with or without evasion technique. However, it is always important to check the traffic and not only rely on Snort.

For Metasploit, Snort relies more on the rules than the pre-processor. The pre-processor (DCERPC2 and HTTP inspect with the support of Stream5) are here to normalizes the traffic and make information that transits understandable and decipherable for the detection engine to ensure and increase the chances of detection for the rules. Each preprocessor has its purpose but it is important to see all the preprocessor as a whole because each preprocessor depends on the other.

By default, the default configuration of the DCERPC2 preprocessor is sufficient it is especially useful for its ability to defragment DCE/RPC. On the other side, the http inspect pre-processor may need some changes. Indeed, some useful options need to be activated such as multi_slash, iis_unicode, apache_whitespace and so on. It really depends on the structure and the type of the server that use the company (IIS, Apache). The administrator may needs to make some choices to adapt the alert that he wants.

The real weak point is that Snort is unable to detect exploit in an encrypted communication and when the code of exploit are encoded by others language, there is still a risk.

The simple way to avoid to be targeted by an exploit from Metasploit, it is to patch the different software on a network regularly or otherwise create a rule the time that the patch comes.

Snort can be considered as one the best solution to protect a SME because its first advantage is that it is free. Large companies will promote solution-based IPS and SIEM but Snort offers a great flexibility and unparalleled scalability through rules. The only inconvenient is that it requires knowledge and basic configuration requires some modification to be really efficient.

For the future, it could be interesting to remake these experiences but in a testing environment where the hardware is limited and the traffic includes not only the traffic of attacks but also some other traffic such as streaming. Snort will be still able to detect the port scan and the exploit. It could be also interesting to test these evasion techniques against other IDS.

6 References

Boltz, M., Jalava, M., & Walsh, J. (2010). "New Methods and Combinatorics for Bypassing Intrusion Prevention Technologies": Stonesoft. available online: http://storage.pardot.com/1912/77027/content_CTA_Technical1_AET.pdf (accessed on 15/01/12)

Maynor, D. (2007). “Metasploit toolkit for penetration testing, exploit development, and vulnerability research”:Syngress. ISBN-10: 1597490741

Nmap, (2012), “Description of Nmap” Available online: http://nmap.org/docs.html (accessed on 17/05/2012)

Ptacek, T. H. (1998). “Insertion, evasion, and denial of service: Eluding network intrusion detection”: DTIC Document. Available online: http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA391565 (accessed on 15/01/12)

Roesch, M., Green, C., Sourcefire Inc, (2011), “SNORT Users Manual 2.9.2”, available online: http://www.snort.org/assets/166/snort_manual.pdf (accessed on 17/05/2012)

Snort (2012) "Description of Snort", available online: http://www.snort.org/snort