Dismantling Intrusion Prevention Systems

Olli-Pekka Niemi
Stonesoft Corporation
Helsinki, Finland
olli-pekka.niemi@stonesoft.com

Antti Levomäki
Stonesoft Corporation
Helsinki, Finland
antti.levomaki@stonesoft.com

Jukka Manner
Aalto University
Helsinki, Finland
jukka.manner@aalto.fi

ABSTRACT

This paper introduces a serious security problem that people believe has been fixed, but which is still very much existing and evolving, namely evasions. We describe how protocols can still be misused to fool network security devices, such as intrusion prevention systems.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: Security and protection.

General Terms

Security.

Keywords

Network, intrusion prevention, evasion, IDS, IPS.

1. INTRODUCTION

The Internet has evolved from the closed environment of academics to a worldwide network. The resources and services available to anyone at any time are huge. Protecting our digital assets from misuse is a major business and area of active research.

Hosts and services on the Internet can be protected by e.g., firewalls, virus scanners, intrusion detection systems (IDS) and intrusion prevention systems (IPS). Our interest are IPS devices (sometimes called network intrusion detection) that do real-time analysis of the traffic and seek to close and raise alarms about flows that include harmful content, e.g., attacks exploiting remote code execution vulnerabilities on server services such as HTTP and MSRPC.

To make attacks unnoticed by IPS devices, the attacker can use so-called evasion techniques, ways to obfuscate the harmful content. Typical tricks include fragmenting the payload into small packets and sending these packs in disorder [2], desynchronizing protocol streams [3], using insertion attacks [2][5] and using various encodings [4][6][8]. The reason that evasions work is the old robustness principle stated by Jon Postel in RFC793 [1] "be conservative in what you do, be liberal in what you accept from others". When the sender deliberately breaks this principle, he can send data that is eventually accepted by the receiver but not necessarily fully understood by an IPS.

Most work on evasions is ten years old. The standard way to reveal evasions is to use traffic normalization to rebuild an obfuscated data stream. The challenge is that obfuscated data streams can be very complex and resource wise expensive to open up. This has led vendors to do anomaly-based detection with minimum amount of normalization and TCP/IP reassembly. The device vendors seem to trust that all known techniques have been disclosed to date.

In our research we have gone further than the current state of the art. We have devised hundreds of atomic tricks on protocols such as IPv4/v6, TCP, NetBIOS, SMB, MSRPC, HTTP and also HTML encoding. These evasions can be combined to form complex evasions [7]. Furthermore, by doing combinations of these atomic evasions on multiple protocol layers and targeting the evasions into detection wise critical stages, we can build advanced evasions techniques (AET) that can by-pass any IPS on the market today. In this paper and demo, we show the Evader tool built to experiment on various evasions. The tool hides harmful content from IPS systems and injects it to the target host. As a result all commercial and open source IPS devices are penetrated, most of them very easily.

2. EVASION RESEARCH THUS FAR

Research on evasions is being done by the scientific and security communities. One of the first papers about evasions was published by Ptacek and Newsham [2] 14 years ago. Many IPS devices still do not handle properly the mentioned methods. In 1998 Horizon wrote an article to Phrack Magazine [3] about desynchronizing TCP stream. Rain Forest Puppy introduced multiple HTTP evasions in [4] and this work was further refined by Daniel Roelker [6]. An insertion evasion called SealMa was disclosed in Phrack [5] and combinations of evasions were introduced by Gorton and Champion [7]. Caswell and Moore [8] summarized at the time known evasion techniques and introduced some new evasions regarding SMB and MSRPC protocols. Examples of more scientific works to solve the evasion problem include a Sigcomm 2006 paper by Varghese et al. [9], and the work by Watson et al. [10]. ACM CCS has had many papers in the past on the topic, e.g., Shunting [11].

SIGCOMM'12, August 13–17, 2012, Helsinki, Finland.
ACM 978-1-4503-1419-0/12/06.

Despite all this work, there is no thorough and systematic discussion of ways to evade network intrusion detection. Our work focuses on understanding in detail the width and size of the problem. Once this is better understood, we can start devising ways to counter the attacks.

3. THE EVADER TOOL

We have implemented a tool we call Evader. It applies network level evasions to send a payload into a remote host through the IPS device under test (DUT). Evader first sends non-malicious payloads that should not be prevented. This is called the false positive test. If this is successful, the malicious payload will be sent. Depending on the selected malicious payload, the remote system is either crashed or compromised via remote code execution. If this happens we know that the evasion was functional against the DUT.

3.1 Features

Evader contains a multilayer network protocol stack. When sending the payload, Evader can apply multiple evasions on various protocols. If the payload e.g. exploits the Microsoft Server Service Vulnerability MS08-067 [12], the evasions applied can be from IP, TCP, NetBIOS, SMB and MSRPC layers. Evader can divide the connection into several stages and every stage can have its own evasions applied. In theory the Evader can produce every possible obfuscated data stream transmitting the payload, but in practice this cannot be tested since there are hundreds of atomic evasions implemented, and virtually endless amount of combinations and stage permutations.

3.2 Test Results

We have used the tool on several commercial IPS systems. The DUTs have been updated to latest software, firmware and signature level. Almost every DUT had to be tuned as well since they either failed to detect the exploit completely or missed it if any evasion were applied. Almost every DUT can be bypassed even with a tuned policy by previously disclosed evasions and AET is not even actually needed. With carefully devised AET, all tested devices fail.

We run evasion tests using different vulnerabilities. Usually a DUT detects the attack and is able to name it with a CVE reference. Yet, with evasions DUTs do not identify the attack anymore even though it may generate protocol anomaly events. The problem with protocol anomalies is that they commonly generate false positives and cannot be used to drop/terminate connections. An example is the prevention of small TCP segments that some vendors use to block evasions. Table 1 shows simple tests where the recent Remote Desktop Vulnerability [13] was exploited through eight commercial IPS appliances. The exploit was taken from Metasploit and the evasions applied were: no evasion (Base), small TCP segmentation (Seg), reverse order sent TCP segments within the congestion window (Reverse), time-wait evasion where multiple decoy connections are opened and closed using same connection tuple before time-wait counter expires (Time-Wait) and overlapping TCP segments with timestamp manipulation (PAWS).

Table 1 Example of experiments (FAIL: successful evasion)

Vendor	Base	Seg	Reverse	Time-Wait	Paws
A	FAIL	FAIL	FAIL	FAIL	FAIL
B	OK	FAIL	FAIL	FAIL	FAIL
C	OK	FAIL	FAIL	OK	OK
D	OK	OK	FAIL	OK	FAIL
E	OK	OK	OK	OK	FAIL
F	OK	FAIL	FAIL	OK	OK
G	OK	OK	OK	FAIL	OK
H	OK	OK	FAIL	OK	FAIL

4. CONCLUSIONS

We have shown how network intrusion prevention is very much behind the means of the attacker and we expect that evasions are being used all the time. We hope to trigger a serious discussion on how to perform network intrusion prevention and protect the society's digital assets. We plan to write a series of articles on these findings in the future.

5. REFERENCES

[1] J. Postel, Transmission Control Protocol. RFC793, IETF, 1981.

[2] T. Ptacek, T. Newsham, Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, Technical report, Secure Networks Inc., 1998.

[3] Horizon jmcdonal@unf.edu, Defeating Sniffers and Intrusion Detection Systems, Phrack Magazine Volume 8, Issue 54 Dec 25th, 1998, article 10 of 12.

[4] Rain Forest Puppy, A look at whisker's anti-IDS tactics

[5] NIDS Evasion Method named "SeolMa", Phrack Magazine Volume 11, issue 57, Phile 0x03, 2001

[6] Daniel J. Roelker , HTTP IDS Evasions Revisited, 2003, http://docs.idsresearch.org/http_ids_evasions.pdf

[7] A. Gorton and T. Champion, Combining Evasion Techniques to Avoid Network Intrusion Detection Systems, Skaion, 2004.

[8] B. Caswell, H D Moore, Thermoptic Camouflage: Total IDS Evasion, Blackhat, 2006

[9] Varghese, et al., Detecting Evasion Attacks at High Speeds without Reassembly, Sigcomm, 2006.

[10] Watson, et al. Protocol Scrubbing: Network Security Through Transparent Flow Modification, IEEE/ACM TON, vol. 12, no. 2, April, 2004.

[11] Gonzalez, et al., Shunting: A Hardware/Software Architecture for Flexible, High-Performance Network Intrusion Prevention. ACM CCS, 2007.

[12] Microsoft, Vulnerability in Server Service Could Allow Remote Code Execution, 2008, MS08-067

[13] Microsoft Remote Desktop Protocol CVE-2012-0002