Appendix B:
Moonlight Maze Technical Report

Tools, Exploits, and Scripts

What follows is a comprehensive breakdown of the artifacts leveraged throughout a window of Moonlight Maze attacks from 1998-1999 that employed the 'HRTest' relay. The operators would pull down TAR archives with different tools, scripts, and exploits to test out on victim systems. Many archives were deployed through this relay but most contained some combination of the artefacts herein described. The total count is *45 binaries*, comprised of *28 SunOS SPARC binaries* and *17 IRIX MIPS binaries*. These include both custom tools as well as repurposed publicly available source code for tools and exploits, and in some cases combinations of the two. Where redundancies and modifications were obvious, these were noted. Where the source code was identified, a link is provided. Additionally, there are *9 scripts* that the operators would execute on victim machines. These include two exploits and several custom scripts meant to efficiently orchestrate malware already operating on a victim system. Since the operators' modus operandi required them to connect to victim networks to issue commands and exfiltration, many binaries were developed to check tasking files under specific names located in the </var/tmp/> directory. The scripts often place these tasks onto these specific files. They would also efficiently prepare data for exfiltration or prune logs for IPs and hostnames that would list further targets on an intranet or interrelated network.

Note that the identified exploits were largely shared on forums designed to improve security awareness and to enable system administrators to get in front of emerging threats that may not be patched by the manufacturers in a timely fashion. Where exploit authors are identified, this is done without assignation of malice and merely as recognition of their skills and contributions to the security community of their time, regardless of their misuse by the attackers.

The binaries, exploits, and scripts have been intermingled into sets as were seen deployed in-the-wild. This allow for an easier understanding of how these sets operated together. Where redundancies occurred, these are noted in the set overviews.

ETAR1 – SPARC Tool and Exploit Set

Overview

ETAR1 is by far one of the most complete tool sets leveraged by the Moonlight Maze operators. It is also less specific and includes redundancies, such as standalone binaries whose functionality is also folded into other binaries also present within the same archive. It features a wide spread of functionalities including multiple log cleaners, kernel patchers, a tunnel redirector, system information stealer, multiple covert channel backdoors, a sniffer, and an xserver keylogger. The archive also includes a wide swath of exploits largely aimed at privilege escalation.

Tools

Filename	cle
MD5	647d7b711f7b4434145ea30d0ef207b0
Size	9.3KB
File Type	SunOS SPARC Binary
Notes	Usage: ./cle filename template_file Custom log cleaner that takes two parameters, a log filename and a template file. It uses the template file to create a list in memory of any strings within the that match the template and then proceeds to purge those that contain a given string. It serves as a log cleaner when provided with a username to wipe from system logs.

Filename	de
MD5	4bc7ed168fb78f0dc688ee2be20c9703
Size	7.8KB
File Type	SunOS SPARC Binary
Notes	A tunnel redirector mean to stay resident in memory. It opens a socket and loops through packet blocks of size 1460

	(or 0x5b4) in order to redirect them elsewhere. Interesting observations: ● The fork of the fork is referred to as 'Vnuk', a transliteration of the Russian for 'grandchild'. ● Similar tools are used by modern Turla to build bridges deeper within partially segmented networks.

Filename	dt25
MD5	e32f9c0dac812bc7418685fa5dda6329
Size	7.3KB
File Type	SunOS SPARC Binary
Notes	A Kernel Patcher meant to kernel memory dependent on command-line parameters. The filename suggests that this version targets SunOS version 2.5. Interestingly, the program uses an unusual convention of error code responses unconventional for the attackers. This implies the source code was likely taken from elsewhere but has yet to be identified: ● "*1100" -> Not enough arguments ● "*1350" -> Can't open device for reading ● "*1300" -> fseek error ● "*1301" -> fread error ● "*1200" -> Address error ● "*1551" -> Successful patch

Filename	dt26
MD5	7dc4f81ed408ff5a369cca737dff064c
Size	10KB
File Type	SunOS SPARC Binary

Notes	Another kernel memory patcher likely targeting SunOS version 2.6 based on the filename. Similar codebase to <dt25> described above including the error message convention.

Filename	gr (2)
MD5	534a1a3212894cf44d8071bdd96ba738
Size	261 bytes
File Type	Script
Notes	A reconnaissance script to collect system configuration files such as host files including remote authentication databases (**/etc/hosts**, **/etc/hosts.equiv**, **/.rhosts**), a list of servers configured for the management of internet services (**/etc/inetd.conf**), and, of course, user accounts and passwords (**/etc/passwd**, **/etc/shadow**). These are copied to the to the folder where the script is executed. It then creates a series of files with the output of the following commands:

Command	Output
df -kb	Disk space in kilobytes
dmesg	System diagnostic messages
/usr/bin/ifconfig -a	Current configuration for all network interfaces
netstat -r	Display network routing tables
pkginfo	Software packages installed on the system
uname -a	Name and basic system information
ps -eaf	List detailed active process information

Filename	lo
MD5	a3164d2bbc45fb1eef5fde7eb8b245ea

Size	18KB
File Type	SunOS SPARC Binary
Source	http://phrack.org/issues/51/6.html
Notes	Implementation of the Loki 2 ICMP information-tunneling backdoor source code published in Phrack 1996-1997. Though the bulk of the codebase appears to come from the published source code, minor alterations suggest attempts to hide the nature of the backdoor. This particular version no longer features obvious references to '[lokid]' and additionally has strings shortened to remove certain vowels: Original: "lokid: client <%d> requested an all kill" Modified: "clnt <%d> rqstd n ll kll" Command-line options have been slightly modified to start with a bang:

Original	**Replaced**
"/stat" /* Stat the client */	!stat
"/swapt" /* Swap protocols */	!swapt
"/quit" /* Quit the client */	!quit
"/quit all" /* Kill all clients and server */	!quit all

Filename	lopg
MD5	9ab532cd3c16b66d98e0e738ddbe05a1
Size	40KB
File Type	SunOS SPARC Binary
Source	http://phrack.org/issues/51/6.html
Notes	As the name suggests, <lopg> is a combination of the LOKI2 backdoor with put/get functionality. It has also been merged with the functionality of <slok>, a strange tool described that watches for commands and interacts with the pine mail client. Just as <lo> described above, it no longer features overt references to [lokid] and has many original strings shortened to remove vowels. The put/get functionality has the following usage: `Usage: @<get/put> <IP> <PORT> <file>` By allowing for direct placement and retrieval/exfiltration of files, the attackers can build a parallel network between infected machines no longer reliant on protocols like FTP and telnet that are likely to be monitored in an ongoing investigation. Constant spelling mistakes and clunky use of the English language suggest the get/put functionality was developed by the operators themselves: ● "ERROR: *Can not* open socket...." ● "open file *for* read" ● "open file *for* write" ● "**receving** message" ● "Error in **parametrs**:" ● "ERROR: *Not connect*..." ● "*Connect successful*...."

Filename	slok
MD5	d0f208486c90384117172796dc07f256
Size	8.4KB
File Type	SunOS SPARC Binary
Notes	Custom tool that goes resident when executed and watches ‘/var/tmp/task’ for commands. Depending on the commands received via the task file, it spawns the pine mail reader in an xterm window. Its purpose is unclear.

Filename	snc (2)
MD5	99a4a154ddecffdab5f0bf91f8bfabb8
Size	5.1KB
File Type	SunOS SPARC Binary
Notes	A tool to run a root shell (/bin/sh) by forcing setuid 0 and setgid 0 or setuid to a user defined value. This is useful if you can get a root user to mark the binary as suid, then you can store it in a hidden place on the machine as an easy way to escalate privileges.

Filename	spl
MD5	b4755c24e6a84e447c96b29ca6ed8633
Size	6.1KB
File Type	SunOS SPARC Binary
Notes	Tool that extracts the first and last 1KB from a filename given as argument. The data is written in two files with their names composed as the original filename to which “.head” and “.tail” are appended.

Filename	tdn
MD5	927426b558888ad680829bd34b0ad0e7
Size	91KB
File Type	SunOS SPARC Binary
Notes	Tool is build from tcpdump and libpcap sources. Attackers used tcpdump v 1.118 and compiled the tool in 1996. The tool takes the parameters from an external file named “/var/tmp/task”. The parameters are standard libpcap filter expressions, such as “(port 21) or (port 23) or (port 513)”. The attackers deployed these rules by writing them from shell scripts to “/var/tmp/task”. Accompanying configuration script is described below.

Filename	ts (2)
MD5	7a0d6b2fdc43b1b2a96b6409d4eed6e4
Size	74 bytes
File Type	Script
Content	echo "(port 21) or (port 23) or (port 513) or (port 110)" > /var/tmp/task
Notes	Configuration script for the <tdn> sniffer, meant to place ports into a tasking file checked by <tdn> on startup.

Filename	u
MD5	d98796dcda1443a37b124dbdc041fe3b
Size	9KB

File Type	SunOS SPARC Binary
Source	Based on publicly available tool “utclean.c”: http://cd.textfiles.com/cuteskunk/Unix-Hacking-Exploits/utclean.c

Notes

usage: %s <username> <fixthings> [hostname]

Tool used to clean various system logs such as:

/var/adm/wtmp
/var/adm/utmp
/var/adm/lastlog
/var/adm/wtmpx
/var/adm/utmpx

At the end, it will execute the following commands:

```
ls -la /var/adm/wtmp* ; /bin/cp
./wtmp.tmp /var/adm/wtmp ; rm
./wtmp.tmp
/bin/cp  ./wtmpx.tmp
/var/adm/wtmpx ; rm  ./wtmpx.tmp
ls -la /var/adm/wtmp*
```

It finishes by printing the following two messages:
- “fixthings: done.”
- “Hiding complit...n”

Attackers changed the string “...that's it. peace man :)” to “Hiding complit...n”.

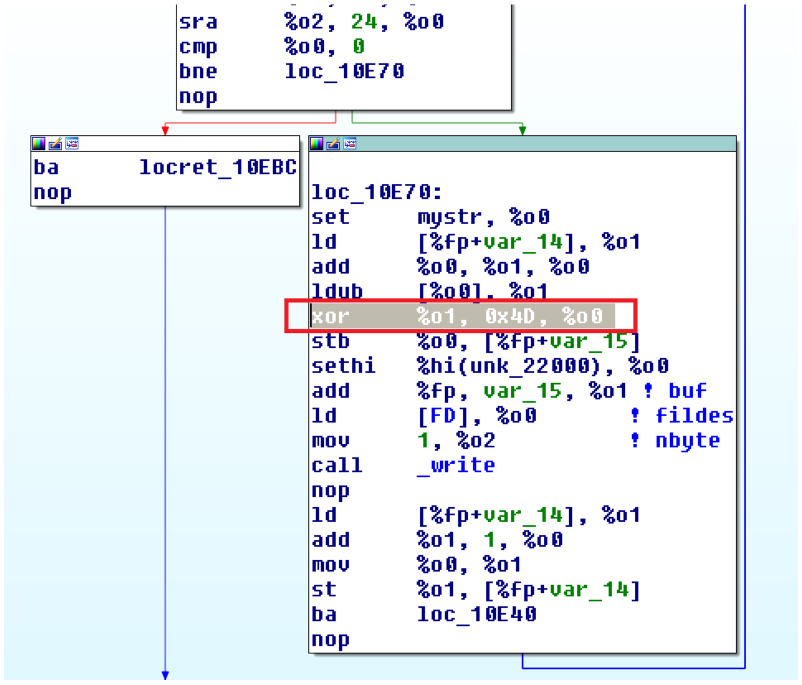
Filename	xk
MD5	4065d2a24240426f6e9912a22bbfbab5
Source	http://web.mit.edu/jhawk/src/xkey.c

Notes

Keylogger for XServer, partially based on the keylogger source code above but modified to write the logs into a log file. It is also made configurable by use of files in '/var/tmp' as is the operators' convention.

It forks to remain resident, then opens '/var/tmp/task' and proceeds to run a series of checks on additional files like '/var/tmp/taskhost', '/var/tmp/tasklog', '/var/tmp/taskpid', and '/var/tmp/taskgid/'. If these are empty, it procures the information and forks.

The log starts at the following routine:



```
sra     %o2, 24, %o0
cmp     %o0, 0
bne     loc_10E70
nop
```

```
ba      locret_10EBC
nop
```

```
loc_10E70:
set     mystr, %o0
ld      [%fp+var_14], %o1
add     %o0, %o1, %o0
ldub    [%o0], %o1
xor     %o1, 0x4D, %o0
stb     %o0, [%fp+var_15]
sethi   %hi(unk_22000), %o0
add     %fp, var_15, %o1 ! buf
ld      [FD], %o0        ! fildes
mov     1, %o2           ! nbyte
call    _write
nop
ld      [%fp+var_14], %o1
add     %o1, 1, %o0
mov     %o0, %o1
st      %o1, [%fp+var_14]
ba      loc_10E40
nop
```

The logs are XORed with 0x4D and the output is saved into: '/var/tmp/.Xtmp01'. However, these logs are found stored in the archives under names like "RES.xk", "A", "B", "ABC" followed by a number, suggesting further manipulation down the line. Decrypted logs show victim communications.

Interestingly, <xk> shares the same log string convention as the <ora> sniffer in the STDN set.

Exploits

Filename	p9
MD5	2213867345a51ecf09d3a747046af78c
Size	6.2KB
File Type	SunOS SPARC Binary
Source	http://borax.polux-hosting.com/madchat/reseau/advisoriez/sun/sunos/5.6/ping.c
Notes	Exploits a buffer overflow in the SunOS ping program used to privilege escalation as root.

Filename	rdi
MD5	34c3ea4d6cc814a174579d295bdd028d
Size	25KB
File Type	SunOS SPARC Binary
Source	http://www.securiteam.com/exploits/3V5QFQ0N5S.html
Notes	Exploit against SunOS rdist program for privilege escalation as root.

Filename	ufsr
MD5	07f070302f42219d37419d23ff9df091
Size	5.9KB
File Type	SunOS SPARC Binary

Source	http://seclists.org/bugtraq/1998/Jun/73
Notes	‘ufsrestore’ exploit (Released 1998)

Filename	ux
MD5	b831cbffa1aee70252bb0f6862265cc9
Size	7.4KB
File Type	SunOS SPARC Binary
Source	http://borax.polux-hosting.com/madchat/reseau/advisoriez/sun/sparc/2.4/holeutmp.c
Notes	Removes a user from utmp logs.

SPTAR – Improved SPARC Attack Set

Overview

SPTAR appears to be an improved toolkit, similar in composition to ETAR1. Many of the binaries are the same. Interesting divergences appear to be modified binaries that include new or improved functionality. In some cases, the operators decided to fold in standalone tools into combined binaries. They've also included a new log cleaner.

Tools

Filename	deg
MD5	8b56e8552a74133da4bc5939b5f74243
Size	8.5KB
File Type	SunOS SPARC Binary
Notes	This is an improved tunnel redirector with functionality similar to <de> in the ETAR1 set. The major modification appears to be an attempt to hide in memory.

Filename	lopg (2)
MD5	1980958afffb6a9d5a6c73fc1e2795c2
Size	45KB
File Type	SunOS SPARC Binary
Source	http://phrack.org/issues/51/6.html http://seclists.org/bugtraq/1995/May/171
Notes	Yet another LOKI2 variant, built on top of the improved <lopg> described in the ETAR1 set but with additional functionality. The attackers have folded in 'utmprm', a utmp log cleaning exploit included as a standalone binary in ETAR1 and other sets.

Filename	wp
MD5	e69efc504934551c6a77b525d5343241
Size	11KB
File Type	SunOS SPARC Binary
Source	http://www.afn.org/~afn28925/wipe.c
Notes	USAGE: wipe [u|w|l|a] ...options... Partially based on source Wipe 1.00 by The Crawler System access log cleaner tool, most likely not written by the attackers. It cleans activity logs in the following system files: ● /var/adm/utmp ● /var/adm/utmpx ● /var/adm/wtmp ● /var/adm/wtmpx ● /var/adm/lastlog

Exploits

Filename	eje
MD5	7bc9d8da363091ad57456f8bd5027ab0
Size	4.1KB
File Type	SunOS SPARC Binary
Source	http://insecure.org/sploits/solaris.eject.html
Notes	‘/bin/eject’ buffer overflow for privilege escalation as root.

Filename	ffb
MD5	26143b006710455888e01df9b58e1913
Size	5.8KB
File Type	SunOS SPARC Binary
Source	https://www.exploit-db.com/exploits/19159/
Notes	FFB buffer overflow that allows user to gain root access, discovered by Cristian Schipor

Filename	sc
MD5	f684ecccd69cca88ba8508711f140240
Size	3.4KB
File Type	SunOS SPARC Binary
Notes	A tool to run a root shell (/bin/sh) by forcing setuid 0 and setgid 0 in an attempt to escalate privileges.

LUTAR – An Early Covert Communications Set

Overview

The LUTAR set appears to have a particular focus on stealth and is comprised of an largely unmodified LOKI2 backdoor, a custom ICMP redirector client meant to interact with it, a simple setuid/getuid attempt to get root shell described in SPTAR, and a utmp access log cleaner. Given the lack of modification to the LOKI2 backdoor, this set is likely an early attempt that came before ETAR1 and SPTAR.

Tools

Filename	cli
MD5	f106ab64b0dc773167a82da7635dfe27
Size	10KB
File Type	SunOS SPARC Binary
Notes	<cli> is a custom ICMP redirector client likely meant to interact with LOKI2 samples. Upon execution it requests the following: ● Enter listen port: ● Enter dst port: ● Enter dst redirect address: ● Enter src addr (in packet): ● Enter dst addr (in packet): It appears that some strings were exchanged back and forth during development as a specific string is shared with the <los> and <lopg (2)> LOKI2 samples: ● "ERROR: Can not open socket...." Interestingly, the file contains more than the operators' adorably broken English. This binary includes a compilation bath that reveals the development name as 'spy_cli.c' and the user profile 'iron': `"/disk3/volume_2/users/`**`iron`**`/myprg/`

	`redirect/solaris/icmp_redirect; //opt/SUNWspro/bin/../SC4.2/bin/cc -O -lsocket -lnsl -c` **`spy_cli.c`** `-W0,-xp”`

Filename	lc
MD5	14cce7e641d308c3a177a8abb5457019
Size	14KB
File Type	SunOS SPARC Binary
Source	http://phrack.org/issues/51/6.html
Notes	<lc> is a relatively straightforward compilation of the LOKI2 source code with some modifications such as the inclusion of a log file called ‘loki.log’, bang replacements for commands (as seen in <lo> and <lopg>), and connection strings: ● ----- Connected to: %s at %s ● ----- Disconnected from: %s at %s This is likely an earlier attempt than those witnessed in the ETAR1 and SPTAR sets as the original LOKI2 strings are largely intact.

IMI – IRIX Tool and Exploit Set

Overview

The IMI toolkit is an IRIX focused toolkit comprised of ported tools seen in the ETAR1 archive like <slok> (now <ilok>) and the <lo> variant of LOKI2 (now <loi>). Additionally, a wealth of different exploits are included focused on providing privilege escalation.

Tools

Filename	ilok
MD5	155d251e6e0dabce21ab26bd03487066
Size	18KB
File Type	IRIX MIPS Binary
Notes	IRIX version of <slok> tool described in the ETAR1 set.

Filename	loi
MD5	dabee9a7ea0ddaf900ef1e3e166ffe8a
Size	29KB
File Type	IRIX MIPS Binary
Source	http://phrack.org/issues/51/6.html
Notes	IRIX version of the <lo> variant of the LOKI2 backdoor described in ETAR1.

Filename	snc
MD5	c73bf945587aff7bc7761b16fc85b5d7

Size	12KB
File Type	IRIX MIPS Binary
Notes	A tool to run a root shell (/bin/sh) by forcing setuid 0 and setgid 0 or setuid to a user defined value. This is useful if you can get a root user to mark the binary as suid, then you can store it in a hidden place on the machine as an easy way to escalate privileges.

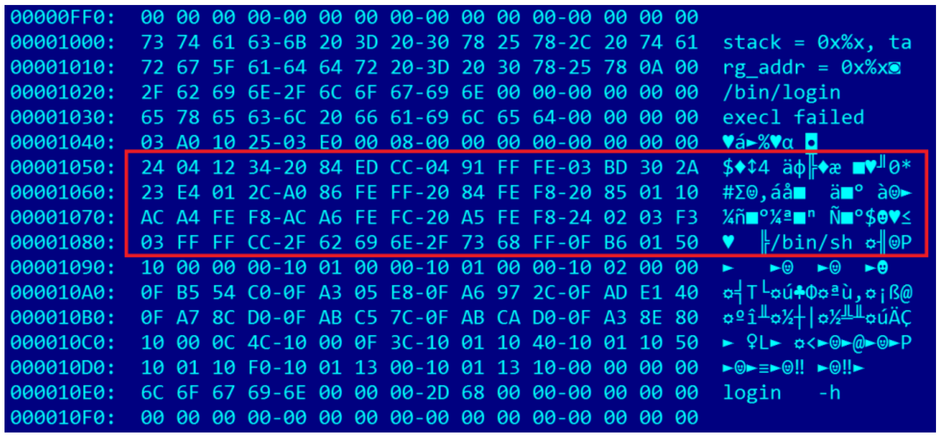
Exploits

Filename	daynotify.sh
MD5	10096abc73b7b7540b607c0ac1a27b49
Size	1.3KB
File Type	Script
Source	http://insecure.org/sploits/IRIX.day5notifier.html
Notes	Copy of a shell executable script written by Mike Neuman in August, 1996 to showcase an exploitable vulnerability in the ‘day5notifier’ program in IRIX 6.2. The attackers copied Neuman’s exact script which allows command execution as root.

Filename	io
MD5	25bcfc394d44d717f20d416354d2126e
Size	176 bytes
File Type	Script
Source	https://www.exploit-db.com/exploits/19163/
Notes	Exploits a vulnerability in the IRIX 6.4 ioconfig binary to allow the execution of arbitrary programs as root Reported by Loneguard and classified as CVE-1999-0314 Exact PoC code shown here with a few comment lines removed

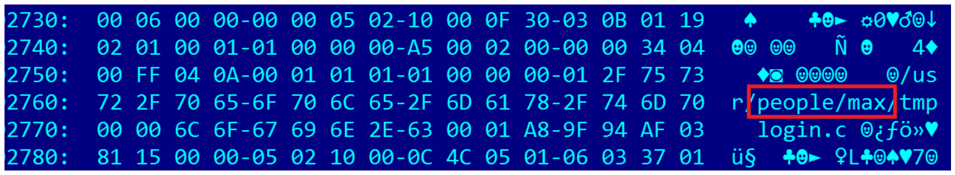
Filename	eject
MD5	864e1d74e610a48c885ac719b5564eb1
Size	17KB
File Type	IRIX MIPS Binary
Source	https://www.exploit-db.com/exploits/19277/
Notes	Irix exploit by Last Stage of Delirium: "A vulnerability exists in the eject program shipped with Irix 6.2 from Silicon Graphics. By supplying a long argument to the eject program, it is possible to overwrite the return address on the stack, and execute arbitrary code as root. Eject is normally used to eject removeable [sic] media from the system, and as such is setuid root to allow for any user at the console to perform eject operations."

Filename	log
MD5	a26bad2b79075f454c83203fa00ed50c
Size	12KB
File Type	IRIX MIPS Binary
Source	http://insecure.org/sploits/IRIX.login.overflow.html
Notes	IRIX v5 and v6 '/bin/login' exploit, originally found by David Hedley <hedley@CS.BRIS.AC.UK> and posted on Bugtraq on 26 May 1997. Shellcode highlighted below:



```
00000FF0:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00001000:  73 74 61 63-6B 20 3D 20-30 78 25 78-2C 20 74 61  stack = 0x%x, ta
00001010:  72 67 5F 61-64 64 72 20-3D 20 30 78-25 78 0A 00  rg_addr = 0x%x◙
00001020:  2F 62 69 6E-2F 6C 6F 67-69 6E 00 00-00 00 00 00  /bin/login
00001030:  65 78 65 63-6C 20 66 61-69 6C 65 64-00 00 00 00  execl failed
00001040:  03 A0 10 25-03 E0 00 08-00 00 00 00-00 00 00 00  ♥á►%♥α ◘
00001050:  24 04 12 34-20 84 ED CC-04 91 FF FE-03 BD 30 2A  $♦↕4 äφ╠♦æ ■♥╜0*
00001060:  23 E4 01 2C-A0 86 FE FF-20 84 FE F8-20 85 01 10  #Σ☺,áå■  ä■° à☺►
00001070:  AC A4 FE F8-AC A6 FE FC-20 A5 FE F8-24 02 03 F3  ¼ñ■°¼ª■ⁿ Ñ■°$☻♥≤
00001080:  03 FF FF CC-2F 62 69 6E-2F 73 68 FF-0F B6 01 50  ♥  ╠/bin/sh ☼╢☺P
00001090:  10 00 00 00-10 01 00 00-10 01 00 00-10 02 00 00  ►   ►☺  ►☺  ►☻
000010A0:  0F B5 54 C0-0F A3 05 E8-0F A6 97 2C-0F AD E1 40  ☼╡T└☼ú♣Φ☼ªù,☼¡ß@
000010B0:  0F A7 8C D0-0F AB C5 7C-0F AB CA D0-0F A3 8E 80  ☼ºî╨☼½┼|☼½╩╨☼úÄÇ
000010C0:  10 00 0C 4C-10 00 0F 3C-10 01 10 40-10 01 10 50  ► ♀L► ☼<►☺►@►☺►P
000010D0:  10 01 10 F0-10 01 13 00-10 01 13 10-00 00 00 00  ►☺►≡►☺‼ ►☺‼►
000010E0:  6C 6F 67 69-6E 00 00 00-2D 68 00 00-00 00 00 00  login   -h
000010F0:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

Tool was compiled by user "**max**". User "**max**" also compiled the "tdn" tool for IRIX.



```
2730:  00 06 00 00-00 00 05 02-10 00 0F 30-03 0B 01 19   ♠    ♣☻► ☼0♥♂☺↓
2740:  02 01 00 01-01 00 00 00-A5 00 02 00-00 00 34 04  ☻☺ ☺☺   Ñ ☻   4♦
2750:  00 FF 04 0A-00 01 01 01-01 00 00 00-01 2F 75 73    ♦◙ ☺☺☺☺   ☺/us
2760:  72 2F 70 65-6F 70 6C 65-2F 6D 61 78-2F 74 6D 70  r/people/max/tmp
2770:  00 00 6C 6F-67 69 6E 2E-63 00 01 A8-9F 94 AF 03    login.c ☺¿ƒö»♥
2780:  81 15 00 00-05 02 10 00-0C 4C 05 01-06 03 37 01  ü§  ♣☻► ♀L♣☺♠♥7☺
```

Filename	pset
MD5	86499f8e6cfc90770a65dc30f1c9939b
Size	17KB
File Type	IRIX MIPS Binary
Source	https://www.exploit-db.com/exploits/19347/
Notes	'/sbin/pset' exploit for IRIX: *"The pset utility, as shipped by SGI with Irix 5.x and 6.x through 6.3, contains a buffer overflow, which can allow any user on the system to execute arbitrary code on the machine as root. Pset is used to configure and administer processor groups in multiprocessor systems. By supplying a well crafted, long buffer as an argument, the return address on the stack is overwritten, allowing an attacker to execute code other than that which was intended."*

Filename	xconsole
MD5	f67fc6e90f05ba13f207c7fdaa8c2cab
Size	13KB
File Type	IRIX MIPS Binary
Source	http://insecure.org/sploits/IRIX.xconsole.cdplayer.xwsh.monpanel.html
Notes	‘/usr/bin/X11/xconsole’ buffer overflow exploit for IRIX by David Hedley.

Filename	xlock
MD5	5937db3896cdd8b0beb3df44e509e136
Size	16KB
File Type	IRIX MIPS Binary
Source	ftp://ftp.ntua.gr/mirror/technotronic/unix/irix-exploits/6.2/xlock.c
Notes	IRIX 6.2 xlock exploit for arbitrary code execution as root.

Filename	xterm
MD5	f4ed5170dcea7e5ba62537d84392b280
Size	13KB
File Type	IRIX MIPS Binary
Source	https://cliplab.org/~alopez/bugs/bugtraq/0307.html
Notes	IRIX exploit for ‘/usr/bin/X11/xterm’ by David Hedley

IMTAR – IRIX Tool and Exploit Set

Overview

Improved IRIX toolkit including the LOKI2 variants with added functionality, additional privilege escalation exploits, and log cleaners.

Tools

Filename	ig
MD5	4110c87e966d4ce6a03c5375353969af
Size	77KB
File Type	IRIX MIPS Binary
Notes	Gzip tool compiled for IRIX, renamed to “ig”

Filename	lo (2)
MD5	f8df359c909ae12f313d9444a6d958d2
Size	41KB
File Type	IRIX MIPS Binary
Source	http://phrack.org/issues/51/6.html
Notes	IRIX port of the <lo> variant of LOKI2, minor differences from <loi>.

Filename	los
MD5	e59f92aadb6505f29a9f368ab803082e

Size	37KB
File Type	IRIX MIPS Binary
Source	http://phrack.org/issues/51/6.html
Notes	IRIX port of the more advanced <lopg> variant of LOKI2 that includes the custom put/get functionality and as well as <slok>/<ilok>.

Filename	ua (or UCL2)
MD5	73a518f0a73ab77033121d4191172820
Size	17KB
File Type	IRIX MIPS Binary
Notes	IRIX log cleaner. English proficiency suggests it was not written by the operators. Strings were ported back to <u> with misspellings.

Exploits

Filename	df3
MD5	008ea82f31f585622353bd47fa1d84be
Size	12KB
File Type	IRIX MIPS Binary
Source	https://www.exploit-db.com/exploits/19274/
Notes	'/bin/df' buffer overflow Irix exploit by David Hedley released in 1997.

Filename	sc (2)
MD5	59198b97f29fcf6e17f8653a99732a74
Size	12KB
File Type	IRIX MIPS Binary
Notes	A tool to run a root shell (/bin/sh) by forcing setuid 0 and setgid 0 in an attempt to escalate privileges.

Filename	ux (2)
MD5	dc9d91e8b2a90df6d25663778a312014
Size	17KB
File Type	IRIX MIPS Binary
Notes	IRIX port of utmprm, which removes a user from utmp logs.

ITDN – IRIX Sniffer Set

Overview

Small set consisting of an IRIX sniffer and its configuration file. The script places the ports to capture onto a file stored at '/var/tmp/task' and checked by <tdni> at startup.

Tools

Filename	tdni
MD5	74af85d293ceb1cfd1a47c0d794e44d5
Size	277KB
File Type	IRIX MIPS Binary
Notes	IRIX port of the <tdn> sniffer tool described in the ETAR1 set. It uses '/var/tmp/task' to read the libpcap rules that will be applied to the network traffic. Configuration script is described below. When compiled, the tool source was located in the following path: `irix:/usr/people/max/mytdn-nsl/tcpdump-3.3/./tcpdump.c`

Filename	ts
MD5	84218bfec08af6a329a277cad9e0044a
Size	60 bytes
File Type	Script
Notes	Script to configure the tdn sniffer to capture telnet, ftp and rlogin packets by port number.

STDN – SPARC Sniffer Set

Overview

Small Solaris sniffer set. Interestingly, this includes both a malfunctioning sniffer (<ora>) and a working version already seen in ETAR1. It's possible that this set came before ETAR1 and was used during a testing phase. Interestingly, despite the issues with <ora>, the developers return to 'solsniffer' to develop <td_tr>, their most improved sniffer.

Tools

Filename	ora
MD5	7b86f40e861705d59f5206c482e1f2a5
Source	http://read.pudn.com/downloads/sourcecode/hack/sniffer/951/solsniffer.c__.htm *Merged with:* http://www.mit.edu/afs.new/athena/astaff/source/src-9.3/third/sysinfo/lib/std/dlpi.c
File Type	SunOS SPARC Binary
Notes	Tool is based on solsniffer, from 1994, created by Michael R. Widner (atreus, J.Galt). It was merged with Neal Nuckolls's (Sun Internet Engineering) DLPI "test" kit. It has built in support to filter out smtp, ftp, rlogin and telnet connections. The main purpose would be the interception of usernames and passwords leaked by the insecure authentication methods from these protocols. It checks for the presence of "/var/tmp/gogo" and will exit without displaying any errors if the file is missing.

Filename	tdn
MD5	927426b558888ad680829bd34b0ad0e7
Size	91KB

File Type	SunOS SPARC Binary
Notes	Sniffer described in detail in the ETAR1 Set where it is leveraged with a different configuration script <ts (2)> Known shell scripts that write to '/var/tmp/task' are <ts>, <ts (2)> and <tsa>.

Filename	tsa
MD5	58e4aa80f14c16e9292bd8f4535fb0cd
Size	74 bytes
File Type	Script
Notes	Script to configure the tdn sniffer to capture telnet, ftp, pop3 and rlogin packets by port number.

STR – Improved SPARC Sniffer Set

Overview

This set is comprised of a vastly improved Solaris sniffer and tools for post-processing the information captured with it. These include scripts to cut out IPs from logs, a utility to resolve those IPs to hostnames, and a script and a gzip binary to prepare files for exfiltration.

Tools

Filename	g
MD5	338f20250b99d8dc064ba7ce8a9f48e1
Source	68KB
File Type	SunOS SPARC Binary
Notes	Compiled version of gzip, renamed to “g”.

Filename	get
MD5	7c930162a676c46ac590342c91402dca
Size	9.5KB
File Type	SunOS SPARC Binary
Notes	Tool which uses gethostbyaddr to get the internet host names for all the IPs specified in an input file. It writes the resolved hostnames into an output file and logs errors to “error.log”.

Filename	gr
MD5	d8347b2e32086bd25d41530849472b8d
Size	342 bytes

File Type	Script
Notes	Shell script to extract all unique source and destination IP addresses from td_tr “RES.u” and “RES.s” logs into the output file “list”.

Filename	td_tr
MD5	66c8fa9569d6b5446eb865544ed67312
Size	187KB
File Type	SunOS SPARC Binary
Source	http://read.pudn.com/downloads/sourcecode/hack/sniffer/951/solsniffer.c__.htm Tcpdump version 1.18
Notes	One of the main sniffer tools used by the attackers. A combination of “ora” and “tdn”, it is compiled from the improved 951 version of <Solsniffer.c>, released in 1994, tcpdump, and libpcap. It captures packets for commonly used insecure authentication protocols (such as ftp, telnet, pop3) and logs these sessions into a text file with the hardcoded name “RES.u”. The tool contains another filename “RES.s” which is not used in the current versions. Text files contain entries such as:

```
-- TCP/IP LOG -- TM: 0 --
 PATH: 128.160.130.141(35787) => 128.160.11.22(110)
 STAT: 0, 13 pkts, 84 bytes [TH_FIN]
 DATA: USER k[redacted]
     :
     : PASS G!d[redacted]
     :
     : STAT
     :
     : QUIT
     :
     :
```

	Due to the proliferation of this sniffer, the hackers essentially created their own archeological trail, by sniffing their own activities as they proxied through infected systems and then proceeding to exfiltrate these records along with what they were actually interested in removing.

Filename	tr
MD5	35f87672e8b7cc4641f01fb4f2efe8c3
File Type	Script
Size	177 bytes
Notes	Shell script which automates several tasks: ● Capture network packets using td_tr ● Extract all unique source and destination IPs using “gr” ● Get corresponding internet host names for all IPs into a filename called “RES” ● Add “RES” to “res.tar” ● Gzip the “res.tar” using compression level -9