Developing Custom PENTEST Tools for an APT Attack Exercise



Dr. Markku-Juhani O. Saarinen FINSE – May 8, 2014



Outline of Talk

- Advanced Persistent Threats & Remote Access Trojans.
- Middle East: Cyber Wargames and Penetration Testing.
- Developing an APT tool.
- Lessons learned: Budget and implementation notes on our custom "tool of the trade", HAGRAT.
- (Live demo in full version of talk).
- Research Directions.



posing One of China's Cybe

NTN

Where this stuff comes from..

Dr. Markku-Juhani O. Saarinen <mjos@item.ntnu.no>

- Developing SSH2 Protocol and its first implementation.
- Nokia Research, Helsinki Univ. Tech (FDF Funded).
- Security Consulting (CISSP-ISSAP, PCI DSS QSA, etc). Middle East (KSA, UAE, Lebanon, Qatar, Kuwait, Bahrain)
- Ph.D. Information Security, Royal Holloway, University of London "Cryptanalysis of Dedicated Hash Functions."
- PI in DARPA-Funded crypto research project, Revere Security Corp, Dallas, TX.
- Security contracting and PENTEST tool development, Help AG (Banking and Government), Doha, Abu Dhabi, Dubai.
- Career culminates at **NTNU** !



Attack Scenario and Payload



Advanced Persistent Threats & Remote Access Tools: 2011

- A March 2011 attack against **RSA Security** dropped a variant of the **Poison Ivy** Remote Access Tool (**RAT**).
- The attack involved an email sent to its employees which carried an Excel file called "2011 Recruitment plan." This file bundled a **zero-day Flash Player exploit**.
- The Security intrusion resulted in the theft of secret keying data related to the company's SecurID two-factor authentication.
- RSA eventually offered to **replace all SecurID tokens** for their customers; approximately 40 million units.



Military Hacking - Research by Mandiant and Paul Rascagnères

- Mandiant claims: APT1 was ran by PLA "Unit 61398" (2nd Bureau of the People's Liberation Army General Staff Department's (GSD) 3rd Department)
- Stole hundreds of terabytes of data from at least 141 organizations. Focuses on compromising organizations in English-speaking countries.
- Operational at least since 2006. Custom tools + Poison Ivy and Gh0st RAT + lots of manpower in Shanghai.

A G20 Campaign by APT12

- Delivered within a Zip archive, no exploit was apparently involved.
- All attacks used the G20 summit as a theme for the bait (St. Petersburg, 5-6 September 2013).
- All the attacks had their malware contact domains pointing to the same host, 23.19.122.231.
- Malicious files used in the attacks belong to the same malware family and behave in the same way.



Evaluating Security against Advanced Persistent Threat: HAGRAT

- A tool for PENTEST or "Red Team" cyberwar exercises.
- A Remote Access Tool (RAT) for Windows 7 and XP, Linuxplatform Command & Control, Multiplatform Op Interface.
- 100% clean-slate development in March-May 2013 by me for my former employer, a Private Security Op in Dubai.
- Not indiscriminately distributed, hence not identified by anti-malware tools. Source available; can be vetted for a PENTEST exercise, not a *backdoored* backdoor!
- Good indicator for estimating the effort required for such cybermunition development.

 NTNT

Requirement Spec (March 2013)

- **1. Remote command-line shell.** Allows the operator to examine the target system and files contained therein.
- 2. Remote program execution. Facilitate operation of "plugin" tools on target system for additional functionality.
- **3. A control terminal.** A remote operator interface that connects to the Command and Control Server component from a remote location.
- **4. File transfer.** Arbitrary file upload / download from the operator system without additional tools or services.
- **5. Communications security.** Strong encryption and authentication of all traffic. Communication link not identifiable by network analyzers.
- **6. Firewall penetration**. HTTP control channel with the Windows system Proxy settings and credentials in order to effectively penetrate through firewalls.
- **7. Alerts**. System can be configured to issue an alert message such as an e-mail when a specific RAT becomes active and the target system can be accessed.
- **8. Automation.** A script system that allows automatic intelligence gathering from target systems.
- **9. Targeted binaries.** Encoding of server address, persistence mechanism, and other configuration information into the RAT binary executable itself.
- **10. Limited persistence.** A persistence mechanism and a "self-destruct" feature which erases the RAT from the target system after a specified date.

$\Box NTNU$

Architecture: Binaries

Windows target executable:

hagr4t.exe A small Windows executable that allows remote control of the target system.

Linux server binaries:

hrccdServer daemon that manages multiple
simultaneous encrypted connections.hrtermOperator control interface ("terminal").hrhtsImplements HTTP tunneling in the server end.

Internal Server components:

hrcommhrxferPairs a terminal session with the desired target.File transfer helper for intel gathering scripts.

$\Box NTNU$

Windows development: hagr4t.exe



Linux development: hrccd

<pre>islow down of the second second</pre>	~			Terminal	1	- + ×	
<pre>134 564 4737 blinker/blinker.c 139 725 vlinker/blinker.c 139 735 4722 blinker/bears.c 139 734 338 blinker/cbtest.c 130 734 338 blinker/cbtest.c 130 734 338 blinker/cbtest.c 130 734 4033 hagritc/hriter.c 130 744 hagritC/hriter.c 130 744 hagritC/hriter.c 130 744 hagritC/hriter.c 130 744 hagritC/hriter.c 130 1053 9774 hagrit/hagrit/hagritLing155], sarg[16]; 121 7/ handle incoming connection 142 10982 hagritC/hriter.c 143 1058 hagritC/hriter.c 144 11e(256], sarg[16]; 145 1145 blinker/cbtean.h 145 1145 blinker/cbtean.h 145 1145 blinker/cbtean.h 145 1145 blinker/cbtean.h 146 1145 linker/cbtean.h 147 1208 1145 blinker/cbtean.h 147 1208 1145 blinker/cbtean.h 147 1208 1145 blinker/cbtean.h 148 1266 hagritC/hr.h 148 126 hagritC/hr.h 148 126 hagritC/hr.h 148 126 hag</pre>	mjos@	dang:~/	Deskto	op/share/hagrat\$ wc */*.c	* ** ** **	*/* • */* • */*/* •	
<pre>199 725 4722 blinker/cbeas.c 19 34 blinker/cbeas.c 19 19 34 blinker/cbeas.c 19 19 19 19 19 19 19 19 19 19 19 19 19 1</pre>	184	564	4757	blinker/blinker.c	Ŧ	hrccd.c (~/Desktop/share/hagrat/hagratcc) - gedit - +	×
<pre>13 34 338 blinker/obtext.c 427 1288 blinker/obtext.c 336 1042 10992 hagratec/hrcnon.c 337 1048 0373 hagratec/hrcnon.c 338 1048 0373 hagratec/hrcnon.c 339 1043 9774 hagrat/hagrat/hagrat.n 339 1043 9777 hagrat/hagrat/hagrat.n 339 1043 9777 hagrat/hagrat/hagrat.n 349 1043 9777 hagrat/hagrat/hagrat.n 349 1043 92518 total 352 151 1145 blinker/obtem.h 55 1151 1145 blinker/obtem.h 55 1151 1145 blinker/obtem.h 56 1143 1266 hagratec/hr.n 57 121 1393 blinker/obtem.h 56 1143 1266 hagratec/hr.n 57 121 1393 blinker/obtem.h 56 1143 1266 hagratec/hr.n 57 121 1393 blinker/obtem.h 56 1145 blinker/obtem.h 57 121 1393 blinker/obtem.h 57 121 1393 blinker/obtem.h 56 133 1266 hagratec/hr.n 57 121 1393 blinker/obtem.h 56 133 1266 hagratec/hr.n 57 121 1393 blinker/obtem.h 57 121 1393 blinker/obtem.h 58 135 136 134 blinker/obtem.h 59 135 135 blinker/obtem.h 59 14 (dinker) 14 (dinker) 14 (dink</pre>	199	725	4722	blinker/cbeam.c	File Edit	View Search Tools Documents Help	
427 1286 11365 hagratec/hrcd.c 336 1408 8973 hagratec/hrtks.c 337 1408 8973 hagratec/hrtks.c 338 1614 hardoch/hardode.c 220 238 1615 614 hardoch/hardode.c 238 1616 6277 hagrat/hagrat.c 239 161 6277 hagrat/hagrat.c 239 161 6277 hagrat/hagrat.c 239 161 16260 *sarg[16]; 241 1435 hagratec/hr.h 220 251 1435 hagratec/hr.h 220 252 143 1436 hagratec/hr.h 220 251 1435 hagratec/hr.h 220 int r. connect(cblnk t *dbs) 252 153 1435 hagratec/hr.h 220 1115 252 153 1436 hagratec/hr.h 220 1116 253 154 143 1266 hagratec/hr.h 220 343 1266 hagratec/hr.h 230 hagratec/hr.hagratec/h	18	34	318	blinker/cbtest.c	E breed		
<pre>336 1042 10982 hagratec/hrms.c 338 1098 973 hagratec/hrms.c 339 1016 974 hagratec/hrms.c 330 1016 927 hagrat/hagrat/magrat.c 330 1016 927 hagrat/hagrat/hagrat.c 330 1016 927 hagrat/hagrat/hagrat.c 331 1015 011kkr/blankr.h 331 1026 hagratec/hr.n 331 1026 hagratec/hr.n 331 1026 hagratec/hr.n 331 1026 hagratec/hr.n 331 1026 hagratec/hr.n 333 1026 hagratec/hagrat5 of hagraf 333 1026 hagratec/hagrat5 of hagraf 334 1026 hagratec/hagrat5 of hagraf 335 1026 hagratec/hagrat5 of hagraf 335 1026 hagratec/hagrat5 of hagraf 336 1026 hagratec/hagrat5 of hagraf 336 1026 hagratec/hagrat5 of hagraf 337 (f (binks, shakel(bs, 10)) 338 1.1. hbinker - c h./blinker/ 339 1.1. hbinker - c h./blinker/ 330 3.1. hbinker - c h./blinker/ 330 3.1. hbinker - c hr/blinker/ 330 3.1. hbinker - c hr/chinker - c - c 331 1 = 0; 1 = 0;</pre>	427	1208	11305	hagratcc/hrccd.c	E mccu.c		
<pre>33 1988 8974 hagratec/hrts.c 188 1986 8974 hagratec/hrts.c 188 1986 8403 hagratec/hrts.c 280 { 181 434 4343 hagratec/hrts.c 282 638 6403 hagrat/hagrat.c 282 647 hagrat/hagrat/hagrat.c 283 1986 8403 hagrat/hagrat.c 284 uintE 1 d[16], bwf[16]; 285 1977 hagrat/hagrat/proxy.c 284 uintE 1 d[16], bwf[16]; 285 1977 hagrat/hagrat/hagrat.c 286 403 hagrat/hagrat/hagrat.h 286 403 1266 hagratec/hr.h 287 197 198 blinker/beam.h 288 1988 9978 hagrat/hagrat.h 288 1988 9978 hagrat/hagrat2 for hagrat/hagrat.h 289 198 1989 92518 total 290 4wll 03 1/blinker/cbeam.c 295 4wll 03 1/blinker - cb.rts.c 295 4wll 03 1/blinker/cbeam.c 295 4wll 03 1/blinker/cbeam.c 296 4wll 03 1/blinker/cbeam.c 297 4wll 04 1/blinker/cbeam.c 298 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 297 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 298 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 299 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 290 4wll 03 1/blinker/cbeam.c 291 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 292 4wll 03 1/blinker/cbeam.c 293 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 293 4wll 04 (fgets(line, sizeof(line) - 1, f) 1= NULL) { 294 4wll 04 (file) file file = file - file</pre>	396	1042	10982	hagratcc/hrconn.c	217 // h	andle incoming connection	
<pre>382 1935 9/44 hagratcc/hrtefm.c 38 85 614 hardcode/hardcode.c 210 sint hr connect(cblnk t *cbs) 221 struct sigaction ss; 222 685 843 hagrat/hagrat/hagrat/magrat.c 223 877 hagrat/hagrat/hagrat/hagrat/hagrat/sigation 224 struct sigaction ss; 225 struct sigaction ss; 226 struct sigaction ss; 227 int i, j, n; 228 struct sigaction ss; 229 struct sigaction ss; 229 struct sigaction ss; 229 struct sigaction ss; 220 struct sigaction ss; 221 int i, j, n; 221 struct sigaction ss; 222 struct sigaction ss; 223 struct sigaction ss; 224 struct sigaction ss; 225 struct sigaction ss; 226 struct sigaction ss; 227 struct sigaction ss; 228 struct sigaction ss; 229 struct sigaction ss; 220 struct sigaction ss; 221 struct sigaction ss; 222 struct sigaction ss; 223 struct sigaction ss; 224 struct sigaction ss; 225 struct sigaction ss; 226 struct sigaction ss; 227 struct sigaction ss; 228 struct sigaction ss; 229 struct sigaction ss; 220 struct sigaction ss; 221 struct sigaction ss; 221 struct sigaction ss; 222 struct sigaction ss; 223 struct sigaction ss; 224 struct sigaction ss; 225 struct sigaction ss; 226 struct sigaction ss; 227 struct sigaction ss; 228 struct sigaction ss; 229 struct sigaction ss; 220 struct sigaction ss; 221 struct sigaction ss; 221 struct sigaction ss; 221 struct sigaction ss; 222 struct sigaction ss; 223 struct sigaction ss; 224 struct sigaction ss; 225 struct sigaction ss; 226 struct sigaction ss; 227 struct sigaction ss; 228 struct sigaction ss; 229 struct sigaction ss; 220 struct sigaction ss; 221 struct sigaction ss; 221 struct sigaction ss; 221 struct sigaction ss; 222 struct sigaction ss; 223 struct sigaction ss; 224 struct sigaction ss; 225 struct sigaction ss; 226 struct sigaction ss; 227 struct sigaction ss; 228 struct sigaction ss; 229 struct sigaction ss; 229 struct sigaction ss; 221 struct sigaction ss; 221 struct sigaction ss; 222 struct sigaction ss; 223 struct sigaction ss; 224 struct sigaction ss; 225 struct sigac</pre>	353	1068	8973	hagratcc/hrhts.c	218		
<pre>188 494 4933 hagratcc/mrter.c 220 { 220 83 614 hardcod/hardccode.c 222 int 1, j, n; 221 61 1145 blinker/blinker. 222 int 1, j, n; 223 111 133 blinker/blinker.h 224 int 2[56]; *srg[16]; 224 131 134 blinker/blinker.h 225 int 1, j, n; 225 151 1145 blinker/blinker.h 226 int 2[6]; *srg[16]; 227 191 1393 blinker/blinker.h 226 int 2[6]; *srg[16]; 228 131 145 blinker/blinker.h 226 int 2[6]; *srg[16]; *srg[16]; 239 113 126 hagratcc/hr.h 226 int 2[6]; *srg[16]; *srg[</pre>	382	1035	9744	hagratcc/hrterm.c	219 int	hr_connect(cblnk_t *cbs)	
<pre>28 89 80 041 MarGeode/MarGeode.c 221 situ(1, j, n; 38 1066 9227 hagr4t/hagr4t/main.c 38 1066 9227 hagr4t/hagr4t/hagrat.c 223 char line[256], *sarg[16]; 224 char line[256], *sarg[16]; 225 char line(r).c 226 char line(r).c 227 char line(r).c 228 char line(r).c 229 char line(r</pre>	168	464	4343	hagratcc/hrxfer.c	220 {		
<pre>222 Bod Bage Bage Page Page Page Page Page Page Page P</pre>	35	85	614	hardcode/hardcode.c	221	struct sigaction sa;	
<pre>33 1900 922 http://magret/hagret</pre>	292	4056	8463	hagr4t/hagr4t/hagrat.c	222	int i, j, n;	
363 1435 1393	330	1012	9227	hagr4t/hagr4t/main.c	223	<pre>char line[256], *sarg[16];</pre>	
<pre>32 131 135 blinker/cbean.h 23 134 blinker/cbean.h 24 143 1266 hagratcc/nr.h 25 151 145 blinker/cbeam.h 26 // register timeouts with alarm() 75 191 133 blinker/cbeam.h 29 sa.sa_handler = hr.alarm_handler; 34 143 1266 hagratcc/nr.h 29 sa.sa_handler = hr.alarm_handler; 39 sa.sa_handler = hr.alarm_handler; 31 alarm(hr_handshake_timeout); // timeout in handshake 39 1039 2518 total 30 sectors/bare/hagrat/hagrat.h 39 1039 2518 total 30 sectors/bare/hagrat/hagrat.h 39 1039 2518 total 31 (Colnk_shake1(cbs, 1d)) 32 cr vall -03 fl./blinker -c ./blinker/237 32 // shake phase 1 (get the client identifier) 32 ffl(f = fopen(hr_secrets_file, "r")) == NULL) 32 cr vall -03 fl./blinker -c hrterm.c - 233 32 ffll06("blnk_shake1(")); 32 cr vall -03 fl./blinker -c hrterm.c - 244 32 ffll06("blnk_shake1(")); 33 cr vall -03 fl./blinker -c hrterm.c - 244 34 ffll06("blnk_shake1(")); 34 cr vall -03 fl./blinker -c hrterm.c - 244 34 ffll06("blnk_shake1(")); 35 cr vall -03 fl./blinker -c hrterm.c - 244 35 for (j = 0, n = 0; line[j] = 0;) 36 cr vall -03 fl./blinker -c hrths/blinker/244 34 ff (line[j] == '#') { // break on comment 35 for (j = 0, n = 0; line[j] = 0;) 35 ffl(j = 0; n = 0; line[j] = 0;) 35</pre>	307	1013	1202	hlinkor/blinkor_b	224	uint8_t id[16], buf[16];	
<pre>24 131 145 011Ker/cbeam.n 54 143 1266 hagratcc/hr.n 75 191 1393 blinker/blinker.h 227 // register timeouts with alarm() memset(&sa, 0, sizeof(sa)); 52 151 145 blinker/cbeam.n 3498 1033 92518 total mjosddang:-/Desktop/share/hagratChagratcc 239 (/ shake phase 1 pr alarm.handler; 240 241 103 1/blinker - c n./blinker 241 245 hagratc/hagratc/hagratcc 242 // shake phase 1 (get the client identifier) 243 (c whall -03 1/blinker - c/blinker 244 // scan our local database for key 245 (r = fopen(hr_secrets_file, "r")) == NULL) 246 (r = fopen(hr_secrets_file); 247 // break on comment 248 if (f = fopen(hr_secrets_file); 249 // split the remaining into parameters 249 // split the remaining into parameters 240 // split the remaining into parameters 241 103 1/blinker - c hrths 243 // split the remaining into parameters 244 // split the remaining into parameters 244 // split the remaining into parameters 245 // break on comment 246 245 247 // break on comment 247 // break on comment 248 249 34 24</pre>	57	151	1145	blinker/blinker.n	225	FILE *f;	
<pre>27 // register timeouts with alarm() 57 191 1393 blinker/blinker/blanker/blinker/beam.h 52 151 145 blinker/blanke</pre>	52	142	1266	biinker/cbeam.n	226		
<pre>memset(&sa, 0, sized(sa); 52 151 145 binker/cbean.h 54 143 1266 hagratcc/hr.h 3498 10339 92518 total mjosddang:-/Desktop/share/hagrats dc hagratcs gcc -Wall -03 -1/binker - c.hrccd.c -0 gcc -Wall -03 -1/binker - c.hrccd.c -0 gcc -wall -03 -1/binker - c.hrced.c -0 gcc -wall -03 -1/binker - c.hrterm.c -0 gcc -wall -03 -1/binker - c.hrter.c -0 gcc -wall -03 -1/binker -0 -0 gcc -wall -03 -1/binker -0 gcc -wall -03 -1/bi</pre>	54	143	1200	high according to the second s	227	// register timeouts with alarm()	
<pre>sa.sa_handler = hr_alam_handler; sa.sa_handler = hr_alam_handler; sas.sa_handler = hr_alam_handler; sas</pre>	52	151	11/5	blinker/cheam b	228	<pre>memset(&sa, 0, sizeof(sa));</pre>	
<pre>sigaction(SIGALKM, &sa, NULL); 75 212 1005 hagr4t/hagr4t/hagr4t.h 3498 10339 92518 total mjosddang:-/Desktop/share/hagr4t/hagr4tCs gcc -Wall -03 -I/blinker -c n./blinker gcc -Wall -03 -I/blinker -c n./blinker/cbeam.o. gcc -Wall -03 -I/blinker -c nrterm.c. gcc -Wall -03 -I/blinker -c nrter.c. gcc -Wall -03 -I/blinker -c nrter.c. gcc</pre>	52	142	1266	bagratec/br_b	229	sa.sa_handler = hr_alarm_handler;	
<pre>alarm(hr_handshake_timeout);</pre>	75	212	1695	hagratt/hagrat/hagrat	230	sigaction(SIGALRM, &sa, NULL);	
<pre>/// shake phase 1 (get the client identifier) mjos@dang:-/Desktop/share/hagrat/hagratcs gcc -Wall -03 -I/blinker -c n./blinker gcc -Wall -03 -I/blinker -c n./blinker gcc -Wall -03 -I/blinker -c hrterm.c -0 gcc -Wall -03 -I/blinker -c hrter.c -0 gcc -Wall -03 -I/blinker -c hrterm.c -0 gcc -Wall -03 -I/blinker -c hrterm.c -0 gcc -Wall -03 -I/blinker -c hrter.c -0 gcc -Wall -03 -I/blinker -c hrterm.c -0 gcc -Wall -03 -I/blinker -c hrter.c -0 gcc -W</pre>	3409	10220	02510	total	231	alarm(hr_handshake_timeout); // timeout in handshake	
<pre>// shake phase 1 (get the client identifier) gcc Wall -03 -1/blinker -c hrcd.c -0 gcc Wall -03 -1/blinker -c/blinker/266 gcc Wall -03 -1/blinker -c hrterm.c -0 gcc -0 hrced hrceno/blinker/cbeam.0 gcc Wall -03 -1/blinker -c hrcmcn.c -0 gcc Wall -03 -1/blinker -c hrcmcn.c -0 gcc -0 hrcen hrceno gcc -0 hrceno gcc -0 hrceno gcc -0 hrcen hrceno gcc -0 hrceno</pre>	5498	10000	Deskt	n/share/hagrats_cd_bagra	232		
<pre>indcodends: //osinter/indcodends/ gcc -Wall -03 -I/blinker -c hrccd.co gcc -wall -03 -I/blinker -c hrterm.c -0 gcc -wall -03 -I/blinker -c hrterm</pre>	miose	lang.	Deskte	on/share/hagrat/hagrates	233	// snake phase 1 (get the client identifier)	
<pre>Hall 03 1/blinker -c/blinker/255 gcc -Wall -03 1/blinker -c/blinker/287 gcc -Wall -03 1/blinker -c hrterm.c -0 gcc -0 hrterm hrterm.o/blinker -c hrterm.c -0 gcc -0 hrconn hrconn.0 gcc -Wall -03 1/blinker -c hrxfer.c -0 gcc -0 hrterm hrters.0 gcc -</pre>	mjos@	dall_0	12 T	/hlinker _c brood c	234	if (cbink_shake1(cbs, id))	
<pre>gcc = wall = 03 = 1/blinker = c/blinker/ gcc = wall = 03 = 1/blinker/cbeam.0 = gcc = wall = 03 = 1/blinker/cbeam.0 = gcc = wall = 03 = 1/blinker = c hrcen = gcc = wall = 03 = 1/blinker = c hrcen = gcc = wall = 03 = 1/blinker = c hrcen = gcc = wall = 03 = 1/blinker = c hrcen = gcc = wall = 03 = 1/blinker = c hrcster.c = gcc = gcc = gcc = gcc = gcc = gcc = phrcsd hrterm hrconn hrxfer hrhts = phrcsd hrterm hrconn hrxfer hrhts = phrcsd hrterm hrconn hrxfer hrhts = cp hrccd hrterm hrconn hrxfer hrts =</pre>	gee I	Jall 0	3 -1.	/blinker - /blinker/	235	FAILOG("cblnk_shake1()");	
<pre>gcc -vall -03 -1/blinker/cbeam.o gcc -vall -03 -I/blinker/cbeam.o gcc -vall -03 -I/blinker -c hrterm.c -0 gcc -o hrcon hrconn.o gcc -vall -03 -I/blinker -c hrtfer.c -0 gcc -o hrxfer hrxfer.o gcc -o hrxfer hrxfer.o gcc -vall -03 -I/blinker -c hrths.c -0 gcc -vall -03 -I/blinker -c hrths/bl 244 10 (fgets(line, sizeof(line) - 1, f) != NULL) { // split the remaining into parameters for (j = 0, n = 0; line[j] != 0;) { // break on comment line[j] =0; // break; // skip spaces line[j++] = 0; C < Tab Width: 4 < Ln 219, Col9 INS</pre>	gee I	Jall 0	3 -1.	/blinker -C/blinker/	236		
<pre>gcc - Wall -03 -I/blinker -c hrterm.c -0 gcc - Wall -03 -I/blinker -c hrcnon.c -0 gcc - Wall -03 -I/blinker -c hrcnon.c -0 gcc - Wall -03 -I/blinker -c hrcnon.c -0 gcc - Wall -03 -I/blinker -c hrcnor.c -0 gcc - whrtser hrcron hrcfer hrts.c -0 gcc - whrtser hrchon hrxfer hrts/bl gcc - whrtser hrtse/bl gcc - whrtser hrtser -0 gcc - whrtser -0 gcc - whrtser hrtser -0 gcc - whrtser hrtser -0 gcc - whrtser -0 gcc</pre>	gee -	hrced	brood	d o /blinker/chosm o	237	// scan our local database for key	
<pre>Gec - ohrterm hrterm.o/blinker/cbeam.o gec - ohrconn hrconn.o gec - ohrconn hrconn.o gec - ohrsfer hrxfer.o gec - ohrsfer hrxfer.o gec - ohrster hrxfer.o gec - ohrster hrxfer c - chrsts.c - o gec - ohrsts hrhs.o gec - ohrster hrecon hrxfer hrhts/bl njos@dang:-/Desktop/share/hagrat/hagratces FAILOG(hr_secrets_file); FAILOG(hr_secrets_file);</pre>	gee -		11 CCC	/hlinker_c_brtorm.c.	238	<pre>if ((f = fopen(hr_secrets_file, "r")) == NULL)</pre>	
<pre>gcc - Wall -03 -1/blinker -c hrxfer.c -0 gcc -0 hrconn hrconn.0 gcc -0 hrtsfr hrxfer.o gcc -0 hrsfer hrxfer.o gcc -0 hrts hrts.0 cc hrcen hrsm hrconn hrxfer hrhts/bi mjos@dang:-/Desktop/share/hagratccs248 249 250 251 252 253 254 253 254 254 255 254 255 254 255 255 255 255</pre>	gee -	hrtor	m brt	arm o /blipkor/choam o	239	FAILOG(hr_secrets_file);	
<pre>gcc - ohrconn.o gcc - ohrconn.o gcc - ohrcher hrxfer.o gcc - ohrsfer hrxfer.o gcc - ohrths hrts.o cp hrccd hrterm hrconn hrxfer hrhts/bi mjos@dang:-/Desktop/share/hagrat/hagratces 242 gcc - ohrths hrts.o cp hrccd hrterm hrconn hrxfer hrhts/bi mjos@dang:-/Desktop/share/hagrat/hagratces 244 245 247 248 249 247 248 249 249 249 249 240 249 240 240 240 240 240 241 244 245 240 241 244 245 240 247 248 249 249 249 250 251 252 253 254 249 254 249 254 254 254 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 254 255 255</pre>	gee -		3 .T	/hlinker -c broon c	240		
<pre>gcc - Wall -03 - I/blinker -c hrxfer.c -o gcc - o hrxfer hrxfer.o gcc - while of size of (line) - 1, f) != NULL) { while (fgets(line, size of (line) - 1, f) != NULL) { // split the remaining into parameters gcc - o hrxfs hrxfs.o cp hrccd hrterm hrconn hrxfer hrhts/bi mjos@dang:-/Desktop/share/hagrat/hagratcs 242 while (fgets(line, size of (line) - 1, f) != NULL) { // split the remaining into parameters for (j = 0, n = 0; line[j] != 0;) { 116 (line[j] == '#') { 116 (line[j] == '#') { 116 (line[j] == '#') { 116 (line[j] = 0; 116 (line[j] = 0; 117 (line[j] = 0; 117 (line[j] = 0; 118 (line[j] = 0; 119 (line[j] = 0; 119 (line[j] = 0; 119 (line[j] = 0; 110 (line[j]</pre>	acc -	hrcon	n bree		241	1 = 0;	
<pre>gcc -o hrxfer hrxfer.o gcc -o hrxfer hrxfer.o gcc -o hrxfs hrhts.o cp hrccd hrterm hrconn hrxfer hrhts/bi mjos@dang:-/Desktop/share/hagrat/hagratccs 246 247 246 247 246 247 248 248 249 248 249 249 249 249 249 249 249 249 249 249</pre>	acc -		13 -T	/hlinker -c bryfer c -c	242	n = 0;	
<pre>gcc -Wall-03 -I./blinker -c hrhts.c -o gcc -o hrhts hrhts.o cp hrccd hrterm hrconn hrxfer hrhts/bi mjos@dang:-/Desktop/share/hagrat/hagratcc5 248 249 248 249 248 249 248 249 251 252 253 254 248 249 253 254 248 249 253 254 248 249 253 254 254 255 254 255 254 255 255 255 255</pre>	acc -	hryfo	r hrvi	fer o	243	wnile (Tgets(line, sizeof(line) - 1, †) != NULL) {	
<pre>gcc -o hrits intro. c minister o gcc -o hrits intro. gc -o hrits</pre>	acc -		3 -T	/hlinker -c brbts c -o	244	//lik kka generation inka generation	
gde of windth mindtold 245 cp hrock hrtern broon hrxfer hrts /bi mjos@dang:~/Desktop/share/hagrat/hagratccs 248 249 if (line[j] == '#') { // break on comment 1ine[j] = 0; break; 252 3 if (isspace(line[j])) { // skip spaces 1ine[j++] = 0; C ▼ Tab Width: 4 ▼ Ln 219, Col 9 INS 4	acc -	hrhte	hrht	s o	245	// split the remaining into parameters	
<pre>op model m commune where min () = 0, n = 0; line[]] != 0;) { mjos@dang:~/Desktop/share/hagrat/hagratccs 24 24 24 24 24 24 24 24 24 24 24 24 24</pre>	cn hr	cd hrt	erm b	rconn bryfer brbts /bi	246		
<pre>if (line[j] == '#') { // break on comment line[j] = 0; break; 252 } 253 if (lsspace(line[j])) { // skip spaces line[j++] = 0; C Tab Width: 4 Ln 219, Col 9 INS </pre>	mios®	lang:~/	Deskte	on/share/hagrat/hagratees	247	ror (] = 0, n = 0; line[]] != 0;) {	
<pre>249 if (Ine[j] == '#') {</pre>	mjos@	ang.	DESKL	op share/ hagrat/ hagrates	248		
250 line[j] = 0; 251 break; 252 } 253 if (isspace(line[j])) { // skip spaces 254 line[j++] = 0; C ▼ Tab Width: 4 ▼ Ln 219, Col 9 INS 4		1	-		249	<pre>ir (ine[j] == '#') { // break on comment</pre>	
251 break; 252 } 253 if (isspace(line[j])) { // skip spaces 254 line[j++] = 0; C ▼ Tab Width: 4 ▼ Ln 219, Col 9 INS 4 C ▼ Tab Width: 4 ▼ Ln 219, Col 9 INS 4					250	<pre>iine[j] = 0;</pre>	
252 253 254 1ine[j++] = 0; C ▼ Tab Width: 4 ▼ Ln 219, Col 9 INS 4 C ▼ Tab Width: 4 ▼ Ln 219, Col 9 INS 4					251	break;	
253 17 (1sspace(11ne[j])) { 77 skip spaces 254 1ine[j++] = 0; C Tab Width: 4 Ln 219, Col 9 INS 4					252		
C Tab Width: 4 Ln 219, Col 9 INS	-	ma!			253	<pre>ir (isspace(line[j])) { // skip spaces</pre>	
C • Tab Width: 4 • Ln 219, Col 9 INS	NY!	100			254	line[]++] = ⊍;	
	-	18-				C 🔻 Tab Width: 4 💌 Ln 219, Col 9 IN:	S
		- P					1
							1 ſ
							· I
							- 1
							- 1

Advanced Firewall Penetration

- Always *outbound* http connection using WININET.DLL. Data flows out encoded in HTTP *requests* and commands go in *responses*.
- Obtains proxy address and authentication information from Internet Explorer and forges user-agent; traffic
- Traffic looks like innocent browsing with Internet Explorer.
- Arbitrary TCP tunneling within a HTTP (port 80). File transfers and remote desktop all wrapped into the HTTP requests.
- Based on ME experience out from 95% of end-user corporate / governmental firewalls, even those with proxy authentication.
- Strong proprietary authentication and encryption, not using system libraries for this. Not detected by current IDS systems.
- A relay mechanism implemented on other end which bounces the traffic to/from the real C2 server.

Demo: Setup (I'll give a demo in full version)





HAGRAT is in Many Ways Superior to PLA RATs

- Military-class communications security: Early versions of Blinker & CBEAM encryption developed for this project.
- The hagr4t.exe binary size is only 12kB ! Note that stuxnet and flame were in megabyte range.
- Excellent firewall penetration even through authenticating web proxies. Fakes a browser connection rather than proprietary port & protocol.
- Robust Linux Command and Control (r00tbsd took over PLA's Poison Ivy CC in his hackback).
- Not as easily detectable due to discrimination in usage.



Conclusions and Further Work: Cybermunitions on a Budget

- Our RAT has only about 3500 lines of code. Three months and a US \$30,000 budget was required (\approx 1 JDAM).
- Safe: Can be used in red team exercises (or simulated cyber warfare campaigns) against live targets.
- Droppers available from metaspoit etc. (However APT12 intelligence gathering efforts w.r.t. St. Petersburg G20 meeting did not even use a zero-day.)
- Easy, cheap and fast ? However I have 20 years of coding and 15 years of PENTEST / Ethical Hacking experience + a PhD in Cryptanalysis.
- Operators need only moderate technical skills (e.g. CISSP), more tenacity and social engineering required.

$\Box NTNU$

Thank you! Questions?

Developing a Grey Hat C2 and RAT for APT Security Training and Assessment

Markku-Juhani O. Saarinen * <mjos@cmdctrl.cc>

CMDCTRL.CC

Abstract. A Remote Access Tool/Trojan (RAT) is a program that allows an external (malicious) operator to invisibly control a host. The operator may examine the system contents, transfer files, and run tools such key- and network sniffers to gain further access. RATs are often inserted on targets by forged e-mails or by

GreHack '13 writeup at: http://www.mjos.fi

