Developing Custom PENTEST Tools for an APT Attack Exercise

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

NTNU

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.

NTNU

Where this stuff comes from..

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

1997 Developing SSH2 Protocol and its first implementation.

2000 Nokia Research, Helsinki Univ. Tech (FDF Funded).

2004 Security Consulting (CISSP-ISSAP, PCI DSS QSA, etc). Middle East (KSA, UAE, Lebanon, Qatar, Kuwait, Bahrain)

2009 Ph.D. Information Security, Royal Holloway, University of London "Cryptanalysis of Dedicated Hash Functions."

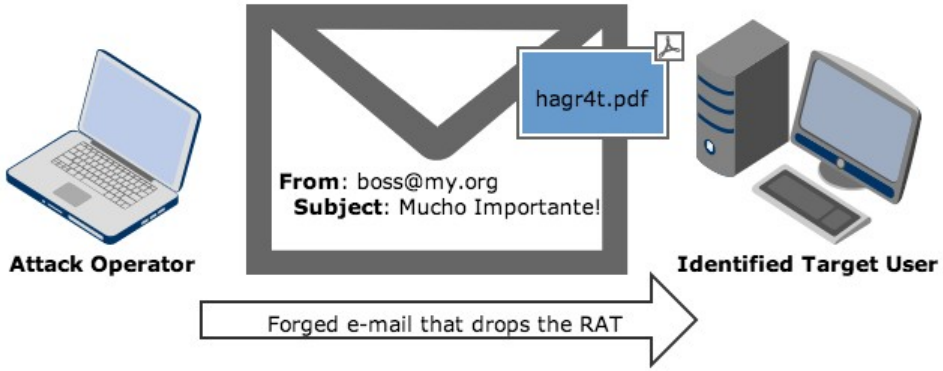
2010 PI in DARPA-Funded crypto research project, Revere Security Corp, Dallas, TX.

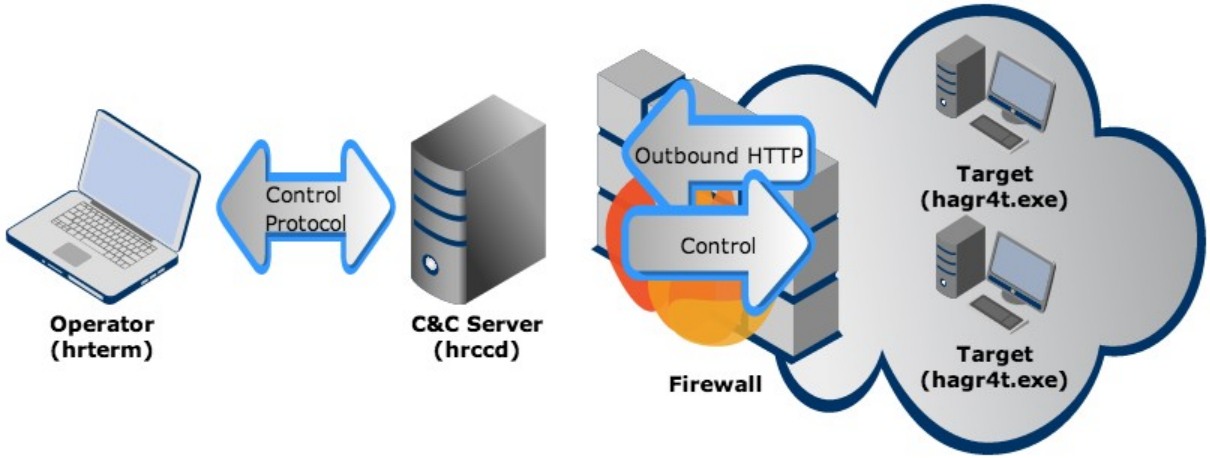
2013 Security contracting and PENTEST tool development, Help AG (Banking and Government), Doha, Abu Dhabi, Dubai.

2014 Career culminates at **NTNU** !

NTNU

Attack Scenario and Payload





NTNU

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.

NTNU

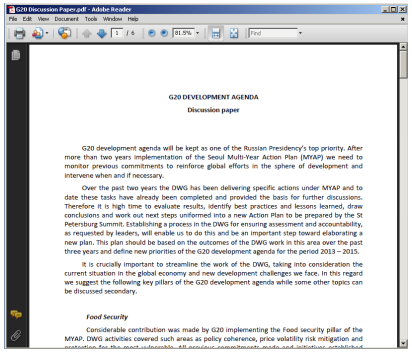
Military Hacking – Research by Mandiant and Paul Rascagnères

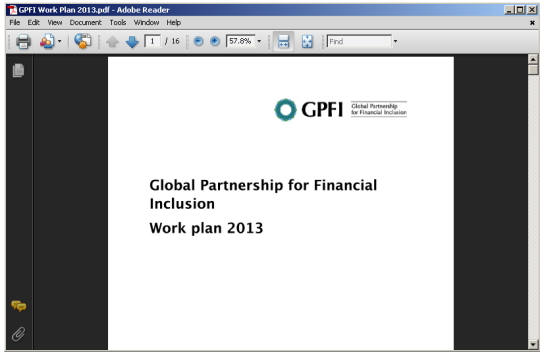
- Mandiant claims: APT1 was ran by PLA "**Unit 61398"** (2nd Bureau of the People's Liberation Army General Staff Department's (GSD) 3rd Department)
- Stole hundreds of terabytes of data from at least 141 organizations. Focuses on compromising organizations in English-speaking countries.
- Operational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

NTNU

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.





NTNU

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, Linux-platform 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.

NTNU

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.

NTNU

Architecture: Binaries

Windows target executable:

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

Linux server binaries:

hrccd Server daemon that manages multiple simultaneous encrypted connections.

hrterm Operator control interface ("terminal").

hrhts Implements HTTP tunneling in the server end.

Internal Server components:

hrcomm Pairs a terminal session with the desired target.

hrxfer File transfer helper for intel gathering scripts.

NTNU

Windows development: hagr4t.exe

hagr4t - Microsoft Visual Studio Express 2012 for Windows Desktop

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP

Local Windows Debugger | Release | Win32

blinker.c | proxy.c | main.c | hagrat.h | hagrat.c

(Global Scope) | hagrat(cblnk_t * cc)

```c
// hagrat.c
// 22-Mar-13  Markku-Juhani O. Saarinen <mjos@cmdctrl.cc>
// hagrat protocol code

// ****************************************************************
// ** CONFIDENTIAL RESEARCH CODE (c) HELP AG. DO NOT DISTRIBUTE **
// ****************************************************************

#include "hagrat.h"

// global transfer buf
char xfer[HR_XFERSIZ];

// send an error message down the pipe
void cblnk_send_c2serr(cblnk_t *cbs, DWORD err, const char *title)
{
    char *msg = NULL;
    char buf[256];

    FormatMessageA(FORMAT_MESSAGE_ALLOCATE_BUFFER |
        FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
        NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPSTR) &msg, 0, NULL);
    if (msg != NULL) {
        sprintf(buf, "%s: (%d) %s\n", title, err, msg);
        LocalFree(msg);
    } else {
        sprintf(buf, "%s: (%d) Unknown Error\n", err, title);
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'hagr4t' (1 project)
  - hagr4t
    - Debug
    - Release
    - blinker.c
    - blinker.h
    - cbeam.c
    - cbeam.h
    - hagrat.c
    - hagrat.h
    - main.c

Solution Explorer | Team Explorer | Class View

Output

```
------ Rebuild All started: Project: hagr4t, Co
  blinker.c
  cbeam.c
  hagrat.c
  main.c
  proxy.c
  Generating code
  Finished generating code
  hagr4t.vcxproj -> Z:\share\hagrat\hagr4t\Rel
```

Error List | Output | Properties

Ready | Ln 138 | Col 19 | Ch 19 | INS

NTNU

Linux development: hrccd

Terminal

```
mjos@dang:~/Desktop/share/hagrat$ wc */*.c
  184   564  4757 blinker/blinker.c
  199   725  4722 blinker/cbeam.c
   18    34   318 blinker/cbtest.c
  427  1208 11305 hagratcc/hrccd.c
  396  1042 10982 hagratcc/hrconn.c
  353  1068  8973 hagratcc/hrhts.c
  382  1035  9744 hagratcc/hrterm.c
  168   464  4343 hagratcc/hrxfer.c
   35    85   614 hardcode/hardcode.c
  292   863  8463 hagr4t/hagr4t/hagrat.c
  336  1056  9227 hagr4t/hagr4t/main.c
  307  1013  9777 hagr4t/hagr4t/proxy.c
   57   191  1393 blinker/blinker.h
   52   151  1145 blinker/cbeam.h
   54   143  1266 hagratcc/hr.h
   57   191  1393 blinker/blinker.h
   52   151  1145 blinker/cbeam.h
   54   143  1266 hagratcc/hr.h
   75   212  1685 hagr4t/hagr4t/hagrat.h
 3498 10339 92518 total
mjos@dang:~/Desktop/share/hagrat$ cd hagra
mjos@dang:~/Desktop/share/hagrat/hagratcc$
gcc -Wall -O3 -I../blinker  -c hrccd.c -o
gcc -Wall -O3 -I../blinker  -c ../blinker/
gcc -Wall -O3 -I../blinker  -c ../blinker/
gcc -o hrccd hrccd.o ../blinker/cbeam.o ..
gcc -Wall -O3 -I../blinker  -c hrterm.c -o
gcc -o hrterm hrterm.o ../blinker/cbeam.o
gcc -Wall -O3 -I../blinker  -c hrconn.c -o
gcc -o hrconn hrconn.o
gcc -Wall -O3 -I../blinker  -c hrxfer.c -o
gcc -o hrxfer hrxfer.o
gcc -Wall -O3 -I../blinker  -c hrhts.c -o
gcc -o hrhts hrhts.o
cp hrccd hrterm hrconn hrxfer hrhts  ../bi
mjos@dang:~/Desktop/share/hagrat/hagratcc$
```

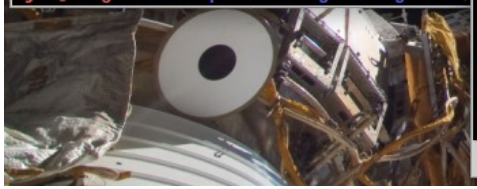
hrccd.c (~/Desktop/share/hagrat/hagratcc) - gedit

File Edit View Search Tools Documents Help

hrccd.c ×

```c
// handle incoming connection

int hr_connect(cblnk_t *cbs)
{
    struct sigaction sa;
    int i, j, n;
    char line[256], *sarg[16];
    uint8_t id[16], buf[16];
    FILE *f;

    // register timeouts with alarm()
    memset(&sa, 0, sizeof(sa));
    sa.sa_handler = hr_alarm_handler;
    sigaction(SIGALRM, &sa, NULL);
    alarm(hr_handshake_timeout);                // timeout in handshake

    // shake phase 1 (get the client identifier)
    if (cblnk_shake1(cbs, id))
        FAILOG("cblnk_shake1()");

    // scan our local database for key
    if ((f = fopen(hr_secrets_file, "r")) == NULL)
        FAILOG(hr_secrets_file);

    i = 0;
    n = 0;
    while (fgets(line, sizeof(line) - 1, f) != NULL) {

        // split the remaining into parameters

        for (j = 0, n = 0; line[j] != 0;) {

            if (line[j] == '#') {               // break on comment
                line[j] = 0;
                break;
            }
            if (isspace(line[j])) {             // skip spaces
                line[j++] = 0;
```

C Tab Width: 4 Ln 219, Col 9 INS



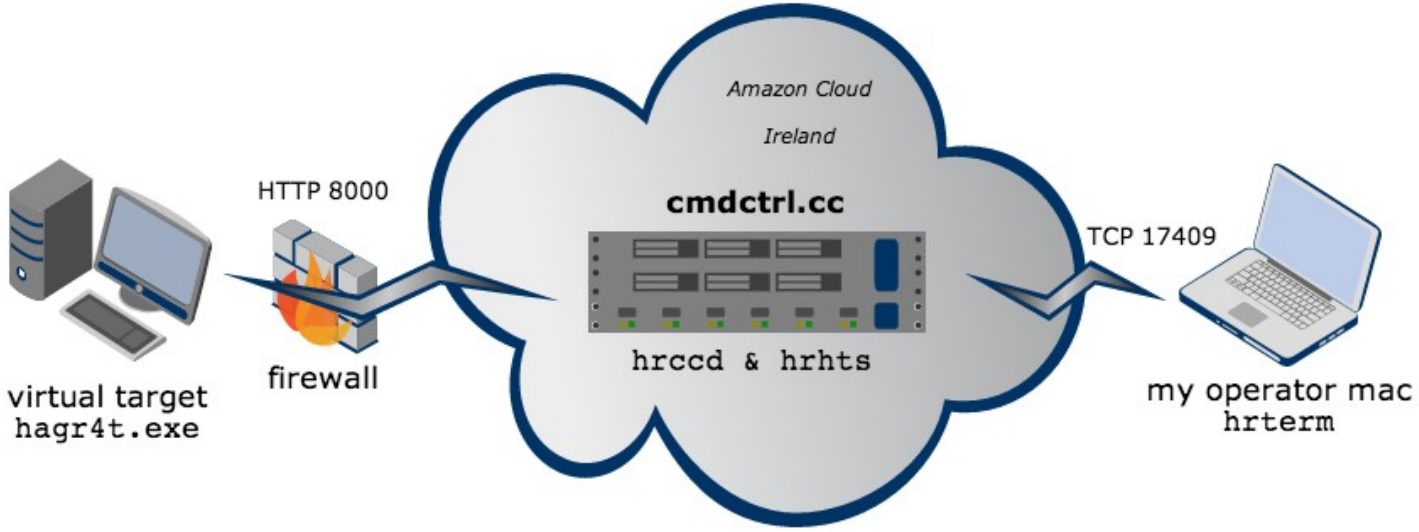
NTNU

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.

NTNU

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



NTNU

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.

NTNU

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 (≈ 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.

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

NTNU