TOP SECRET//SI//REL TO USA, FVEY

(U) S3285/InternProjects

TOP SECRET//SI//REL TO USA, FVEY
From WikiInfo
< S3285
Jump to: navigation, search

(TS//SI//REL) This page contains ideas about possible future projects for the Persistence Division.

Contents

[edit] (TS//SI//REL) CNA Team POLITERAIN

(TS//SI//REL FVEY) TAO/ATO Persistence POLITERAIN(CNA) team is looking for interns who want to break things. We are tasked to remotely degrade or destroy opponent computers, routers, servers and network enabled devices by attacking the hardware using low-level programming. It would be expected that our interns would learn to:

- (U//FOUO) Write drivers for LINUX, Windows, Solaris, or Apple OS.
- (U//FOUO) Use SVN in a group environment.
- (TS//SI//REL) Reverse engineer embedded systems
- (TS//SI//REL) Deliver code that conforms to Op-sec and deniability requirements.
- (TS//SI//REL) Recover equipment that has been attacked.
- (U//FOUO) Work with multiple SME’s to build something unique.
- (TS//SI//REL) Developing an attacker's mind set.

(TS//SI//REL FVEY) POLITERAIN always has a backlog of smaller attacks than those listed below that need to be productized. We are also always open for ideas but our focus is on firmware, BIOS, BUS or driver level attacks. The projects below an intern could be expected to produce results in 4-6 months. Most of the projects are unique enough to allow for results to be briefed or published in a classified venue.

[edit] (U) PASSIONATEPOLKA

(TS//SI//REL FVEY)We have discovered a way that may be able to remotely brick network cards. We need someone to perform research and develop a deployable tool. Intern would have access to driver level developers, mentors

& SMEs but would own the project and be responsible for it.

[edit] (U) ARGYLEALIEN

(TS//SI//REL FVEY)There is a security feature built into many modern hard-drives that allows for zeroization. We want to use this feature to cause the loss of data. Intern would be working closely with a POLITERAIN engineer to develop a solution that would work on multiple vendors leveraging Persistence divisions SMEs, investigating via experimentation. Intern would learn about hard-drive recovery, reverse engineering, hard-drive architecture and much more.

[edit] (U) BARNFIRE

(TS//SI//REL FVEY)This attack effort will erase the BIOS on a brand of servers that act as a backbone to many rival governments. An intern working on this project would need to be a *nix expert with experience with low-level development experience of multiple types and reverse engineering experience. Intern would be working and learning from SMEs during this development. First tour interns would not be considered for this project.

[edit] (U) Hard Drive Recovery

(TS//SI//REL FVEY) When someone really needs the information off of a damaged hard drive, they call Persistence. This would be a unique tour learning from a world-class expert how to fix hardware and firmware problems. You would be working on targets, extracting data, troubleshooting hardware, rebuilding SCSI arrays, and using analytical engineering skill to produce real collection. Position requires good hearing for some of the troubleshooting. Intern should know how to solder.

[edit] (U) IRATEMONK

[edit] (TS//SI//REL) SSD Support

(TS//SI//REL) Integrate SSD research into IRATEMONK products. This will involve 4 different parts:

- (TS//SI//REL) Leveraging research to create ARM-based SSD implant. This works involves reverse engineering SSD firmware and creating C and ARM assembly code to place inside of a firmware image to implement the IRATEMONK algorithm.
- (TS//SI//REL) Create version of the IMBIOS code that supports the SSD implant. This code runs on the x86 host and involves writing both C and

x86 assembly. This work will involve interacting with the firmware implant as well as the code that IMBIOS bootstraps (SIERRAMIST).
- (TS//SI//REL) Add support for the SSD to WICKEDVICAR. WICKEDVICAR is the remote tool used to perform remote survey and installation. This code is C++ and will involve interacting with the firmware implant from a Windows OS.
- (TS//SI//REL) Add the SSD vendor support to the IRATEMONK firmware and implant database tool. This code is mostly python code that interacts with a drive via a Linux driver.

(TS//SI//REL) The SSD support for IRATEMONK project currently offers the greatest variety of new work that an intern might be able to do.

[edit] (TS//SI//REL) Covert Storage Product

(TS//SI//REL) Create a covert storage product that is enabled from a hard drive firmware modification. The idea would be to modify the firmware of a particular hard drive so that it normally only recognizes say half of its available space. It would report this size back to the operating system and not provide any way to access the additional space. The firmware would have a special hook inside of it that on receipt of some custom ATA command, it would "unlock" the rest of the drive on the next boot of the drive. When covert storage is locked, only 1 partition would be present on the drive. When unlocked, the firmware would fix up the partition table to account for the second hidden partition whose space is now available on the drive. When finished with covert storage, a special command can be sent back to the drive that will lock the drive again. On the next boot, the firmware will hide the extra space and fix up the partition table so only 1 partition exists.

[edit] (U//FOUO) SADDLEBACK

(TS//SI//REL) Utilizing a hard drive's serial port, create a firmware implant that has the ability to pass to and from an implant running in the operating system. In practice, the serial port will be connected to a short hop radio that can communicate with another radio in a system. Doing a firmware modification eliminates the need to tap the SATA bus as was done on other versions of SADDLEBACK. Performing firmware modification will allow for a smaller SADDLEBACK in the form of a laptop drive as opposed to the current version which only comes in a a 3.5 inch version.

[edit] (U//FOUO) ALTEREDCARBON Support

(TS//SI//REL) Develop IRATEMONK implants for the newest Seagate drives including their hybrid drive products. This work will primarily be a reverse engineering effort, but if successful will require updates to both IMBIOS (x86 code, C and assembly), WICKEDVICAR (x86, C++), and SPITEFULANGEL

(python).

[edit] (U//FOUO) FAKEDOUBT Support

(TS//SI//REL) Create an IRATEMONK implementation for ARM-based Hitachi drives. This includes a firmware implant, IMBIOS code, and WICKEDVICAR and SPITEFULANGEL support.

[edit] (U//FOUO) PLUCKHAGEN Support

(TS//SI//REL) Create an IRATEMONK implementation for ARM-based Fujitsu drives. This includes a firmware implant, IMBIOS code, and WICKEDVICAR and SPITEFULANGEL support.

[edit] (U//FOUO) EASYKRAKEN Support

(TS//SI//REL) Add more drive support for ARM-based Samsung drives.

[edit] (TS//SI//REL) USB Hard Drive Persistence

(TS//SI//REL) Develop a capability to install a hard drive implant on a USB hard drive. Since external hard drives are not normally boot from, the new implant will need to be an improved version of MADBISHOP so the hard drive implant will have the ability to manipulate the file system of the drive inside of the firmware itself. Development would consist of 3 main development areas:

- (TS//SI//REL) Reliable, robust, and portable NTFS C code. Other file systems could also be looked into such as FAT, EXT2, etc.
- (TS//SI//REL) Hard drive implant
- (TS//SI//REL) Remote installation over USB

[edit] (TS//SI//REL) IRATEMONK on Server/RAID Systems

(TS//SI//REL) Investigate the feasibility of developing a hard drive persistence implant for Server/RAID systems. This will primarily involve investigating what ATA commands are allowed through various RAID controllers. Also, coming up with a scheme to handle different RAID configurations will be needed since the data on the drives will differ depending on the the RAID setup. It will be important to examine how RAID controllers interact the drives and where data (especially the MBR) is stored. If feasible, this investigation could lead to the extension of IRATEMONK-type

hard drive implants into server spaces which provide increased covert storage and the capability to keep persistence logs which currently aren't available on BIOS only techniques.

[edit] (U//FOUO) Enhancment of testing capabilities

(TS//SI//REL) IRATEMONK makes use of ROGUESAMURAI and actual hardware to perform its testing. The downside of testing with actual hardware is that testing takes many hours/days to complete. Someone with the right level of test development experience can help to improve the current IRATEMONK testing strategy.

[edit] (TS//SI//REL) Self-Encrypting-Drive (SED) Persistence

(TS//SI//REL) SED drives provide additional security measures which often thwart IRATEMONK developer efforts to modify the firmware on these drives. I highly skilled intern with reverse engineering skills and understanding of security in computing systems would be invaluable in tackling one of the persistence divisions more difficult problems.

[edit] (TS//SI//REL) OS Execution

(TS//SI//REL) While a lot of work in the Persistence Division involves modifying firmware, there is still a large need for OS kernel and user-mode expertise. The firmware modification done at the lowest levels of hardware needs a way to obtain execution inside of a running OS so that a DNT payload can be either given execution or installed.

[edit] (U//FOUO) CASTLECRASHER

(TS//SI//REL) CASTLECRASHER is the primary technique used in executing DNT Windows payloads from all payload persistence techniques (i.e. IRATEMONK and SIERRAMISTFREE). It is all Windows native mode code built using Visual Studio. CASTLECRASHER has many advanced techniques in it including thread injection and anti-stack backtracing. In many cases, CASTLECRASHER is closer to the DNT style kernel work than it is to traditional Persistence work. While the current version is quite robust, there are several features that need to be added:

- (TS//SI//REL) Currently CASTLECRASHER doesn't work against systems with 360 Safe installed. We need to find a way around this even if it involves using the older Windows service method of execution. This

will more than likely require a refactoring of how the configuration data of CASTLECRASHER is stored.
- (TS//SI//REL) Develop an automated test suite using the Persistence Division's ROGUESAMURAI test framework to provide more robust testing for this important project.

[edit] (TS//SI//REL) Alternate Windows Execution Technique

(TS//SI//REL) Currently, CASTLECRASHER is the only production quality Windows execution technigue that Payload Persistence techniques have. Another mechanism to execute DNT payloads is needed. Most pre-boot Persistence techniques only have the ability to influence an OS through modifications to the target file system. Work needs to be done to investigate other ways to get execution inside of Windows. This work will start looking at other techniques that have been provided to the Persistence Division from other partners. The feasibility of these techniques should be assessed. If feasible, the technique should be productized into a deployable solution.

[edit] (TS//SI//REL) Mac OSX Execution Technique

(TS//SI//REL) Research needs to be done to investigate different ways that a pre-boot Persistence technique that can modify the target file system can get execution inside of OSX. Maybe start-up scripts can be modified or special files can be added that will get executed. In order for Payload Persistence to work against OSX, a execution technique is needed.

[edit] (U) SIERRAMIST/JUMPDOLLAR

[edit] (U) Extensibility Application

(TS//SI//REL) Create a new extensibility application that checks for a file on the file system that it will run to update the SIERRAMIST partition. This application would replace the current one and would have list of configurable file paths to check. This could be written for both SIERRAMIST and JUMPDOLLAR. Create a ROGUESAMURAI test suite to test all aspects of this app.

[edit] (U) MOPNGO Application

(TS//SI//REL) Update the MOPNGO application to remove the buffer overflow issue it has when more than 512 characters are configured. Also, port this application to JUMPDOLLAR. Create a ROGUESAMURAI test suite to test the application. Look into creating unit tests as well.

[edit] (TS//SI//REL) NTFSD DOS Driver

(TS//SI//REL) The MX team has requested an upgrade of their NTFSD DOS driver for their DOS-based thumb drives used in interdiction deployments. The newest FSM should be compiled into a DOS driver for their use. This problem will be solved with the creation of a MKUSB utility for JUMPDOLLAR.

[edit] (U) Full Featured Shell

(TS//SI//REL) A full featured shell should be written (particularly for JUMPDOLLAR). The ability to run scripts and have some sort of flow control logic would be desirable. This could eventually be used by the MX team as a means to deploy implants. They have expressed a desire to have one environment that can work on a Mac, Linux, or a Windows machine. Provided we have the file system support, this could eventually provide them what they want.

[edit] (TS//SI//REL) Windows Registry Read/Write Capability

(TS//SI//REL) Create the ability from SIERRAMIST/JUMPDOLLAR to be able to read and write the Windows registry. This will provide new capability to apps to do a whole host of new things. It may be possible to install VALIDATOR manually instead of relying on its installers. This work may also allow some IT Geo applications as well if we can tweak some keys.

[edit] (TS//SI//REL) EFI Module

(TS//SI//REL) Build an EFI module out of SIERRAMIST/JUMPDOLLAR apps that can be loaded via normal EFI mechanisms including an EFI shell. This is similar to what Sandia can do with their MOUSETRAP implant. This work could provide a new mechanism to achieve persistence and might prove to be easier than current patching techniques.

[edit] (TS//SI//REL) Linux App

(TS//SI//REL) Rewrite the Linux App to be configurable and add logging into it for use with IRATEMONK. Integrate build into BORGERKING. Investigate

whether or not ROGUESAMURAI can be used to test.

[edit] (TS//SI//REL) File System Support

(TS//SI//REL) The following file systems need support by SIERRAMIST/JUMPDOLLAR:

[edit] (TS//SI//REL) BTRFS

(TS//SI//REL) This file system is slated to become the default in Fedora Core 17 or 18. Work can begin on supporting this now, however. This is applicable primarily for JUMPDOLLAR, but maybe SIERRAMIST as well.

[edit] (TS//SI//REL) LVM

(TS//SI//REL) Back port changes of the LVM code in JUMPDOLLAR's EXT4 FSM to SIERRAMIST's EXT3.

[edit] (TS//SI//REL) EXT4

(TS//SI//REL) Back port JUMPDOLLAR's EXT4 FSM to SIERRAMIST.

[edit] (TS//SI//REL) UFS/ZFS

(TS//SI//REL) Create UFS and ZFS FSMs for JUMPDOLLAR.

[edit] (U//FOUO) BERSERKR

(TS//SI//REL) BERSERKR is a persistent backdoor that is implanted into the BIOS and runs from SMM. Although the core of the code is stable, there are always new requirements against which to develop. This includes new network interface card parasitic drivers as well as applications.

(TS//SI//REL) Some notable applications that need development:

- (TS//SI//REL) KIRKBOMB - Windows kernel examination to detect loaded drivers, running processes as well. There is a prototype which works on Windows 7, this needs to work on XP and 2008 including 64-bit systems.
- (TS//SI//REL) SODAPRESSED - Linux application persistence. Given a running installation of Linux, install some application or inject something into memory which will. This currently works on certain versions of Linux without SELinux enabled.

(TS//SI//REL) There may also be requirements in the near future for:

- (TS//SI//REL) BENTWHISTLE - A collection tool that runs from BERSERKR.

(TS//SI//REL) BERSERKR is often looking to expand its target support. A big way this is done is via adding network card support. Currently, BERSERKR does not support any wireless network cards.

[edit] (U//FOUO) GOPHERRAGE

(TS//SI//REL) GOPHERRAGE is a project that seeks to develop a hypervisor implant that would leverage both AMD and Intel's virtualization technology in order to provide both DNT implant persistence capabilities and a persistent back door.

(TS//SI//REL) Develop a hypervisor implant that would leverage both Intel's and AMDs virtualization technology in order to provide both DNT implant persistence capabilities and a persistent back door access. The idea would be similar to what BERSERKR can do from SMM in that it should be able to use "the machine's network interface card (NIC) to communicate independently of the host operating system (OS)". Also, this hypervisor implant should have full read/write access of host memory so it will be possible to change Host OS behavior in ways that could allow code execution, OS injection, system survye, VM break-in, etc.

(TS//SI//REL) GOPHERRAGE is the Persistence Division's pilot program to apply industry best practices and agile development processes to internal projects. To this end, the project is managed via the Scrum process. Test Driven Development (TDD) practices are used as well in an effort to reduce code defects. The project also is looking to incorporate ideas from DNT such as their SCube build environment.

[edit] (U//FOUO) Windows Tools

[edit] (U//FOUO) WISTFULTOLL

(TS//SI//REL) WISTFULTOLL is the premiere target survey tool for Windows that runs on almost all targets automatically. It brings back information about the target system's machine and operating system that is invaluable for both the Persistence Division and analysts enterprise wide. New features need to be added to WISTFULTOLL as well as it being refactored.

[edit] (U//FOU0) CENTRICDUD

(TS//SI//REL) CENTRICDUD is a tool to read and writes bytes in the CMOS.

It needs to be rewritten and productized so that it can be incorporated into a proper UR plug-in. The driver associated with this tool also needs to be redone as it is being flagged by PSPs for unknown reasons. This tool is used both by the BIOS team as well as the IT Geo team.

[edit] (U//FOUO) STYLISHCHAMP

(TS//SI//REL) STYLISHCHAMP is a tool that can create a HPA on a hard drive and then provide raw reads and writes to this area. This tool should incorporate latest TWISTEDKILT code so that it can support SATA drives. This will allow SWAP to be used on newer systems. Currently, only IDE drives are used.

[edit] (U//FOUO) Network Infrastructure

(TS//SI//REL) TORNSTEAK is a persistence solution for two firewall devices from a particular vendor. We need to port TORNSTEAK from the existing two firewalls to several more from the same vendor. This persistence effort would use one's reverse engineering, computer architecture, "C" programming and assembly language coding skills.

- Read
- Edit
- View history

Actions

- Move
- Watch
- Tag this page

Search

Go Search

Navigation

- Main Page
- Community portal
- Recent changes
- Random page
- Help

Toolbox

- What links here
- Related changes
- Trackback
- Upload file
- Special pages
- Printable version
- Permanent link

social software tools

- JournalNSA
- Tapioca
- Connexions
- LINKUP
- SpySpace
- Round Table
- WikiInfo-NF

partner wikis

- Intellipedia
- CSEC wiki

- GCHQ wiki
- DSD wiki
- GCSB wiki