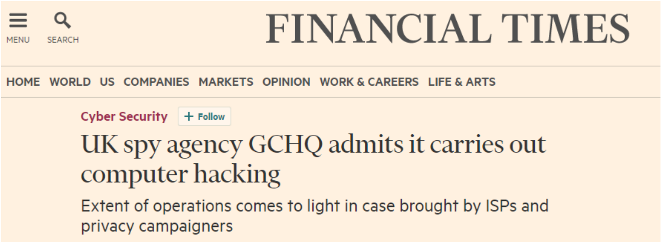
SSD Architectures to Ensure Security and Performance

Jon Haswell

Sr. Director, SSD Firmware Development
Micron Technology, Inc.
Milpitas, California USA

Caught in the Act: Nation State Cyber Espionage



MENU
SEARCH
FINANCIAL TIMES
HOME WORLD US COMPANIES MARKETS OPINION WORK & CAREERS LIFE & ARTS
Cyber Security
+ Follow

UK spy agency GCHQ admits it carries out computer hacking

Extent of operations comes to light in case brought by ISPs and privacy campaigners

Revealed in court case brought by Privacy International

- Actions were dubbed legal within the UK

How do we know ?

- Analysis of malware by anti-virus and security companies
  - Frequently in industry collaborations
- Key companies in this analysis all have acknowledged or suspected links to intelligence agencies
  - Seem to reveal malware from 'the other side'

- Equation Group
  - Has existed for at least 14 years
    - Targeted 42 countries including Iran, Russia, Indian subcontinent, China
  - Most sophisticated malware ever seen
    - Multiple OS/platforms
    - Air gapped networks
- Axion Group
  - Targeted Fortune 500 companies, journalists, NGOs
  - Large command and control network
  - Wide variety of malware
    - Customized for long term persistence
- Lazarus Group
  - Active since 2009
    - Attacked numerous US and S Korean websites
  - Behind Sony Pictures and other destructive attacks
  - Tied to 45 different malware families

The NSA Organization

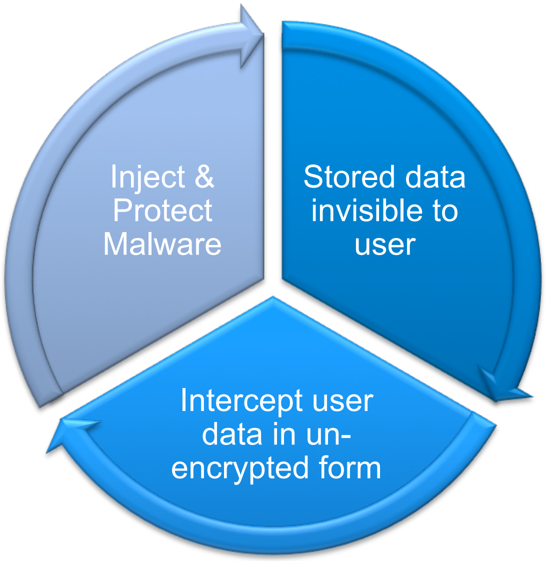
AN EXAMPLE OF AVAILABLE RESOURCES

- Tailored Access Operations (TAO)
  - A cyber-warfare & intelligence-gathering unit of the National Security Agency (NSA).
    - It has been active since at least circa 1998.
    - Identifies, monitors, infiltrates, and gathers intelligence on computer systems being used by entities foreign to the United States.
  - Reportedly has over 1000 employees at central operations at Fort Meade, MD
  - 4 Branches
    - Data Network Technologies: spyware development
    - Telecommunications Network Technologies: network and computer hacking methods
    - Mission Infrastructure Technologies : operates the malware
    - Access Technologies Operations: Physical planting of eavesdropping devices worldwide
      - Reportedly includes personnel seconded by the CIA and the FBI
      - Suspected submarines used to wiretap fiber optic cables around the globe.

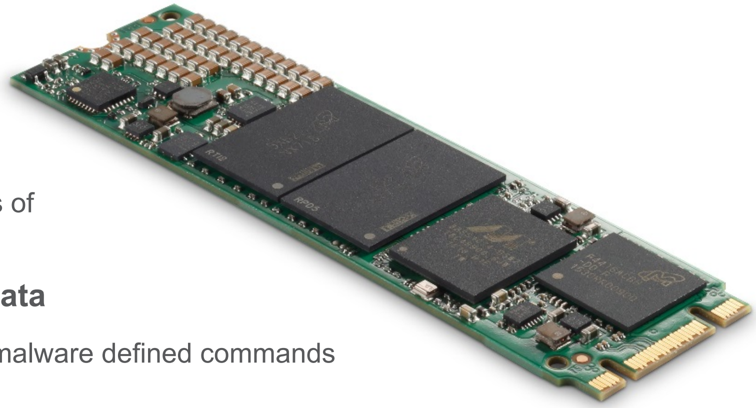
Cyber Warfare/Malware Objectives

Find a route to install malware on target system and ensure it is not discovered or removed

- Avoiding discovery is a key objective
  - Information retrieved by malware loses its value if the target knows the information is compromised
  - Intercept user data in un-encrypted form
- All encrypted data is decrypted when it is needed to be used/ reviewed
  - Targeting data at this time is usually easier than cracking encryption
- Plan B – steal the encryption keys and decrypt at leisure
  - Store data invisibly to user
  - Can be used to hide the malware or temporarily store data prior to it’s clandestine transmission off the system


Inject & Protect Malware
Stored data invisible to user
Intercept user data in un-encrypted form

Attacking the Storage Device



- **Can run sophisticated firmware**
  - Typically has ample processing power and memory for storing and running firmware modified to include malware components
- **Can control the boot sequence of the entire system**
  - Can modify OS booting sequence by presenting alternate images of boot records and other device driver layer code
- **Can provide clandestine storage space for malware and data**
  - Device firmware can create hidden partitions accessible only by malware defined commands to act as temporary storage space
- **Can protect malware from scanners and anti-virus tools**
  - Host based scanners can only scan SSD space presented to host system as part of partitioned volume, firmware controls what is visible in this space
- **Can hide malware from detection**
  - Current anti-virus tools cannot remove malware from drive firmware

Storage Devices are Hackable?

HOW HARD IS IT TO HACK THE FIRMWARE

It can be expensive and time consuming …

- Can I get the spec for the SSD controller?
  - Bribery or social engineering works nicely here – if you really need the spec
- Can I get copies of the firmware to patch or disassemble?
  - Available in unencrypted binary form for download for most devices
  - Reverse compilers, particularly for ARM, available
- To modify SSD firmware does not require a complete understanding of the device
  - Only identification of key routines and entry points

… but has been demonstrated as a practical attack

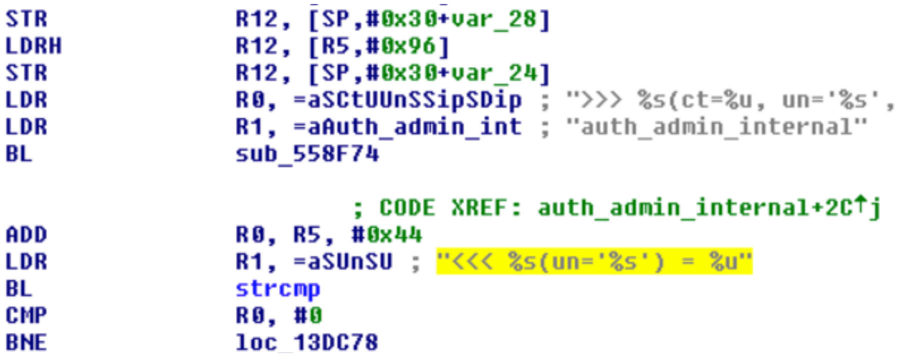
BadUSD- Proof of concept attack

- Security researchers reverse engineered USB storage stick firmware
- Created modified firmware & replacement firmware
  - Hidden partitions
  - Password bypass
- Techniques and firmware published

How to Attack Firmware I

CAUGHT IN DEVELOPMENT: AN “INSIDER ATTACK”

- Actual malware injected into carrier class network switch
  - Coded to look like a *printf* string to avoid detection by source scanning tools
  - Not only enables access but elevates any account to admin privileges



```
STR             R12, [SP,#0x30+var_28]
LDRH            R12, [R5,#0x96]
STR             R12, [SP,#0x30+var_24]
LDR             R0, =aSCtUUnSSipSDip ; ">>> %s(ct=%u, un='%s',
LDR             R1, =aAuth_admin_int ; "auth_admin_internal"
BL              sub_558F74

                        ; CODE XREF: auth_admin_internal+2C↑j
ADD             R0, R5, #0x44
LDR             R1, =aSUnSU ; "<<< %s(un='%s') = %u"
BL              strcmp
CMP             R0, #0
BNE             loc_13DC78
```

- Other fruitful approaches as demonstrated in VPNs
  - Weakening random number generates/Encryption algorithms
    - Enables trivial cracking

How to Attack Firmware II

ATTACKING THE SUPPLY CHAIN

- "Man-in-the-Middle" intercepts device in manufacturing or shipping
    - Carefully open packages
    - Replace firmware
    - Repackage and ship
  – Can occur within manufacturing facility, during shipment and delivery
- Documented instances of such attacks have occurred on high end networking switches and other IT systems

How to attack firmware III

REMOTE DOWNLOAD IN THE FIELD

Equation Group Framework - injected rogue firmware into SSD/HDD

- Provided control of boot sequence, storage of malware, temporary storage of unencrypted user data
- Utilized unprotected vendor unique commands
- Went to extra-ordinary length to avoid rogue firmware being recovered

Rogue master boot record

- First software executed at boot
- Patches windows loader

Windows boots under malware control

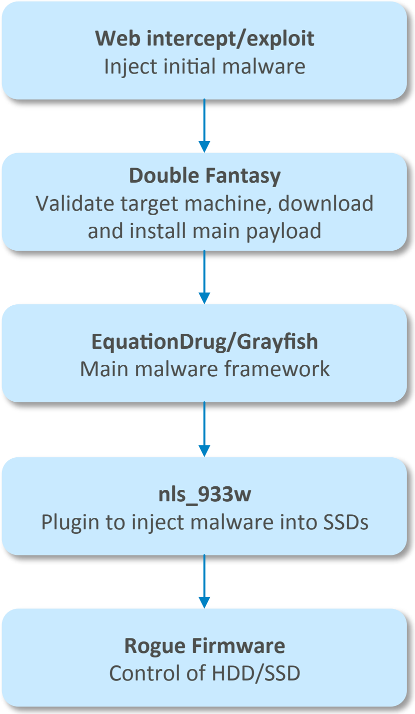
- Hidden control malware loaded into Windows as it boots

During normal operation

- Malware captures documents
- Stores in hidden area of storage device

Later

- Transfers hidden documents to command and control servers



Web intercept/exploit
Inject initial malware

Double Fantasy
Validate target machine, download and install main payload

EquationDrug/Grayfish
Main malware framework

nls_933w
Plugin to inject malware into SSDs

Rogue Firmware
Control of HDD/SSD

How to Attack Firmware IV

CRACK OR BYPASSING ENCRYPTION

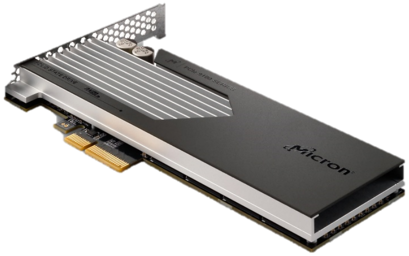
- Encryption is key to securing firmware and user data
  - TCG encryption of user data
  - Signing of firmware images
  - Protecting internal device debug functionality
- Encryption is only valuable when
  - The keys are adequately protected
  - The algorithms are strong and verified
  - Backdoors in the system do not allow it to be bypassed
- Alleged hack on manufacturer of SIM cards containing encryption keys
  - Utilized social engineering and targeting of key employees
    - Allegedly penetrated network
  - Alleged to have obtained billions of private encryption keys
- Weakened random number
  - During standardization process
  - Incorporated into firewalls/VPN gateways
  - Reportedly provided ability to intercept all encrypted traffic with ‘reasonable’ effort

Countering Attacks

The Objective: Protect the SSD at all Phases from any imaginable attack

- Deploy Basic Security Features
- Drive integrity during design and development
- Manufacturing and deployment without interference
- Protection in the field remote attacks and non destructive physical attacks

Basic Security Features



Start with the Self-encrypting Drive (SED)

- Client: TCG Opal/IEEE-1667
- Enterprise: TCG Enterprise

TCG Opal/Enterprise & IEEE 1167 creates a secure encryption environment for *"Data-at-Rest"*

SSD Design Validation Challenges

- Is the firmware actually encrypting?
- Is the firmware intercepting the data before it is encrypted on write, after decryption on read?

- **FIPS 140-2 Level 2 is typical for Storage Devices.**

Level 1: Certification of encryption engine and associated firmware

Level 2: Tamper evident seals to protect access; role-based authentication requirements

Level 3: Tamper resistant casing; tamper response may include zeroing of all critical security parameters (CSP); identity-based authentication requirements

Level 4: Robust tamper resistance and intrusion response; compulsory zeroing of CSPs on intrusion detection; hardened casing for unanticipated environmental conditions

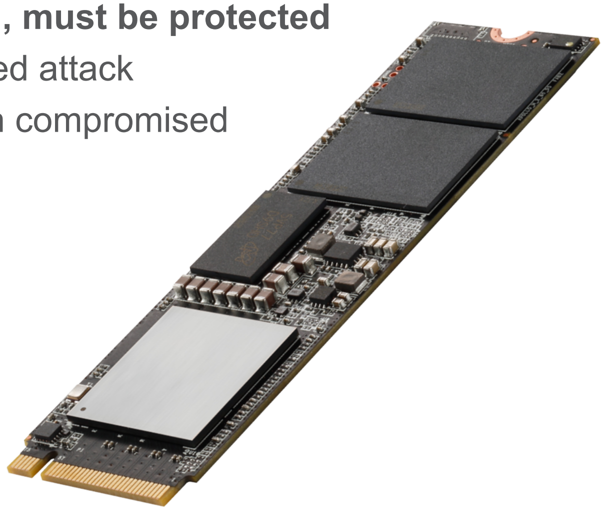
Additional Security Features

- **Digitally Signed Firmware Binaries**
  - Cryptographically signed with signing keys closely controlled by the vendor

- **All vendor unique commands or other abilities, including for debug, must be protected**
  - Mechanisms must be of a suitable strength to withstand sophisticated attack
  - Including replay attacks and possibility host machine used has been compromised

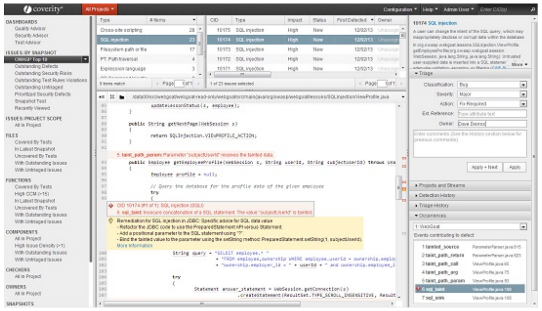
- **Security versioning**
  - The ability to track the security level of firmware/components
  - Prevent the replacement of a component to downgrade security

- **Security logging**
  - The logging of all operations, changes or errors relating to security
  - Primarily for audit purposes



Protection During Design and Development

- **Protection against malicious activity; insider attacks**
  - Injection of deliberate errors of back doors in source code
  - Enforcement of multiple reviewers of all code reduces risk
    - With a full audit trail
  - Exhaustive penetration and fuzz testing to back it up



Type	# Items
Cross-site scripting	28
SQL injection	23
Filesystem path or file	17
PT: Path traversal	4
Expression language	3

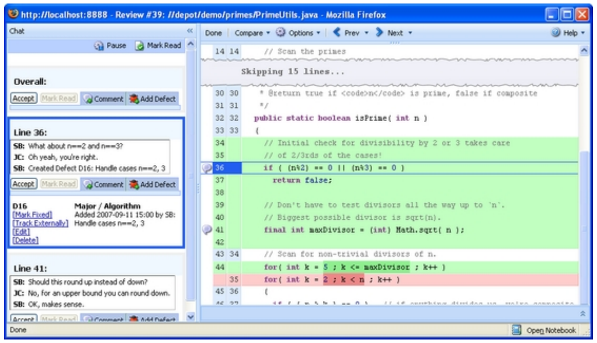
CID	Type	Impact	Status	First Detected
10175	SQL injection	High	New	12/02/13
10174	SQL injection	High	New	12/02/13
10173	SQL injection	High	New	12/02/13
10172	SQL injection	High	New	12/02/13
10171	SQL injection	High	New	12/02/13

10174 SQL injection

A user can change the intent of the SQL query, which may inappropriately disclose or corrupt data within the database

```
        updateLessonStatus(s, employee);
    }

    public String getNextPage(WebSession s)
    {
        return SQLInjection.VIEWPROFILE_ACTION;
    }

5. taint_path_param: Parameter "subjectUserId" receives the tainted data.
    public Employee getEmployeeProfile(WebSession s, String userId, String subjectUserId) throws Una
    {
        Employee profile = null;

        // Query the database for the profile data of the given employee
        try
        {
CID 10174 (#1 of 1): SQL injection (SQLI)
6. sql_taint: Insecure concatenation of a SQL statement. The value "subjectUserId" is tainted.
Remediation for SQL injection in JDBC: Specific advice for SQL data value
- Refactor the JDBC code to use the PreparedStatement API versus Statement.
- Add a positional parameter to the SQL statement using "?".
- Bind the tainted value to the parameter using the setString method: PreparedStatement.setString(1, subjectUserId).
More information
            String query = "SELECT employee.* "
                + "FROM employee,ownership WHERE employee.userid = ownership.employ
                + "ownership.employer_id = " + userId + " and ownership.employee_i

            try
            {
                Statement answer_statement = WebSession.getConnection(s)
                        .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, Result
```

1 tainted_source	ParameterParser.java 615
2 taint_path_return	ParameterParser.java 623
3 taint_path_call	ViewProfile.java 65
4 taint_path_arg	ViewProfile.java 75
5 taint_path_param	ViewProfile.java 93
6 sql_taint	ViewProfile.java 100
7 sql_sink	ViewProfile.java 100



Overall:

Line 36:

SB: What about n==2 and n==3?
JC: Oh yeah, you're right.
SB: Created Defect D16: Handle cases n==2, 3

D16 — **Major / Algorithm**
Added 2007-09-11 15:00 by SB:
Handle cases n==2, 3

Line 41:

SB: Should this round up instead of down?
JC: No, for an upper bound you can round down.
SB: OK, makes sense.

```
14 14     // Scan the primes

     Skipping 15 lines...

30 30    * @return true if <code>n</code> is prime, false if composite
31 31    */
32 32   public static boolean isPrime( int n )
33 33   {
34        // Initial check for divisibility by 2 or 3 takes care
35        // of 2/3rds of the cases!
36        if ( (n%2) == 0 || (n%3) == 0 )
37          return false;
38
39        // Don't have to test divisors all the way up to `n`.
40        // Biggest possible divisor is sqrt(n).
41        final int maxDivisor = (int) Math.sqrt( n );
42
43 34     // Scan for non-trivial divisors of n.
44        for( int k = 5 ; k <= maxDivisor ; k++ )
   35     for( int k = 2 ; k < n ; k++ )
45 36     {
```

- **Protection from inadvertent errors**
  - Applying the same tools as used to protect application software
    - Human designed penetration testing
    - Fuzz testing, subjecting interfaces to random invalid inputs to confirm correct behavior
    - Strict enforcement of coding standards
    - Static analysis

Protecting the Integrity of the Firmware

PUBLIC/PRIVATE KEY CRYPTOGRAPHY AT THE ROOT OF TRUST

To confirm validity of all firmware before execution

To establish trust and allow privileged operations

- Protection of private signing keys is essential
  - **They define who you are and protect the reputation of your company**
  - Use Hardware Security Module (HSM)
    - Ensure private signing keys never leave the protection of a properly hardened HSM
  - Protect the signing infrastructure
    - Physically and logically protect the HSM in a secure data center environment
  - Strict authentication and authorization
    - Ensure the identity of those using private signing keys is validated and that there is strict controls on what material can be signed and when
  - Prepare for security issues
    - Monitor all signing, attempts to sign and network access to signing infrastructure

Manufacturing and Infrastructure

The integrity of the product is only guaranteed if what is designed and tested is what is delivered to the customer

- Security efforts must account for a wide variety of manufacturing environments.
- **Self securing products**
  - Design a product where each component will validate the integrity of other components as the system boots
    - First time and every time, the equivalent of secure boot in a PC within the SSD
  - Can protect itself during manufacturing and delivery to the customer
- **Secured production capacity**
  - Trusted and audited personnel
  - Processes enforcing multiple sets of people to achieve operations

In Summary

Our data are under surreptitious attack like never before

New threats are always appearing. Yesterday’s novel discovery is tomorrows automated attack vector

To protect our customers, and our business, we need to carry forward our reputation for products that can be trusted

Counter measures and strategies to protect our devices are available or can be created