An attack on Apple examined
A deeper dive on attack classes Whitenoise technologies prevent

André Brisson abrisson@wnlabs.com

An attack on Apple examined
A deeper dive on attack classes Whitenoise technologies prevent

The goal of cyber security is to prevent hacks and the consequences that ensue. Traditional cryptography continues to fail.

This paper examines two topics:

1. How did the recent hack on Apple iStores occur and how can we prevent that in the future?
2. How attack classes operate and the SIMPLE logic that detects and prevents them?

One bad Apple

Almost all cyber defenses are post facto meaning that the attack or penetration has already occurred. As an example, let's look at the hack discovered against Apple iStores.

The attack was propagated by code that was inserted into an Apple software developer kit (SDK). Applications were developed using this SDK and then sold or made available in iStores. Reports don't give an indication of how long these applications were sold and the number of devices that are compromised. The Apple breach was malware that stole information and then sent it out. It created a botnet by commandeering the device.

We will discuss a simple way to prevent botnets in a moment but let's quickly look at one of the outcomes of globalization.

A massive amount of electronics destined for western markets are made in foreign countries. US Senator Levine noted the problem long ago and discussed the many compromised microprocessors in US military and government assets.

There are several solutions to solve this. In the interest of brevity, I will provide one.

There is no security without key based cryptography. At time of manufacture, anywhere, these devices can be provided with a generic Whitenoise key. A generic key tells a hacker nothing – every device has the same key and there is no way to distinguish the device other than manufacturer numbers. They do not know the IP address that device will eventually operate on. They will never be able to connect to the device in a future hack because they cannot authenticate.

At time of purchase (in store), or with online activation, the owner of the device chooses a very robust "password or passphrase" or a random number generator is used. That value fundamentally and deterministically changes the generic key on the device to make it unique and private to the device and user. This happens one time. There is nothing the user needs to remember. The system notes the key on this personalization because in a distributed key system the server has copies of all the unique, private keys on its network. After this one-time key provisioning there is never key or offset transmission again.

Note – devices without keys can be provisioned one during one-time online enrollment, authentication and activation.

André Brisson abrisson@wnlabs.com

That answers the manufacturing part of the problem.

The other part of the problem is the botnet that the malware successfully established and which needs to be cleaned up. A valid question is: “Why did it take so long to figure it out?”

The simple logical defense against botnets is discussed first in the next section.

Attack classes

Solutions for cyber security need to be logical and simple so they cannot be broken, so that people can understand why the security works, and so that scalability and interoperability issues do not come into play.

All Whitenoise technologies are driven by a quote by Albert Einstein:

“Make things as simple as possible but no simpler.”

Botnet attack

“A botnet is a number of Internet-connected computers communicating with other similar machines in an effort to complete repetitive tasks and objectives. This can be as mundane as keeping control of an Internet Relay Chat (IRC) channel, or it could be used to send spam email or participate in distributed denial-of-service attacks.” wiki

The malware that creates a botnet is effective only if that malware can send out data from the device to other servers. If data isn’t going out unbeknownst to the victim then no damage can be done.

Simple logic defense

Configure your system so that all data leaving a network is authenticated by a third party server (internal or external) with a different authentication key required for outbound communications. The malware has no knowledge of that key and no ability to learn anything about it because it has never been transmitted and is not at the site that has been penetrated.

If outbound data cannot be authenticated because of no key presence, an incorrect key, an unsynchronized token, or the lack of authorization, the sending device is automatically locked down without human intervention. Failing outbound authentication has served the role of intrusion detection and we immediately know if a device has been compromised or if there is an internal player with bad intentions present. Knowing the device that failed makes remediation and forensic work much simpler – and no damage has occurred.

Distributed denial of service attack (DDoS)

Botnets are generally used to create distributed denial of service attacks on specific targets. We have seen how to prevent them.

“A denial-of-service (DoS) attack is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. A distributed denial-of-service (DDoS) is where the attack source is more than one–and often thousands–of unique IP addresses.” **wiki**

Simple logic defense

Secure session login can be handled by a third party proxy to keep login routines at arms length to the target institution. If there is no key, there is no entry. If there is somehow a pirated key (all the botnets would have that same key) the keys will not be synchronized and access is denied.

The botnet computers trying to attack a network will either have no key or the key will not be synchronized so they can never get onto the network.

It still can be obnoxious (even if not dangerous) and when the proxy notes the activity the network can automatically switch to a mirror site and the compute data regenerated so that the transition to backup sites is seamless. When the event has been handled, traffic can be switched back to the primary site harmlessly.

Man-in-the-Middle attack (M-i-M)

“In cryptography and computer security, a man-in-the-middle attack is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.” wiki

It is IMPOSSIBLE to prevent MiM attacks on public key systems because of how key distribution and traffic is negotiated and conducted.

Simple logic defense

Man-in-the-middle-attacks are not possible in distributed key systems because there is NEVER private key or offset exchange in session. Traffic is moved with authenticated encryption and the MiM attacker has no access to the keys with which they are encrypted so they can do no damage. They can capture encrypted traffic but they cannot open it.

Side Channel attack

"In cryptography, a side-channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. For, example, timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information, which can be exploited to break the system." wiki

Simple logic defense

Side channel attacks are not possible when Whitenoise technologies are used:

The subkey lengths (or registers) are populated with sequential, determinist, random data from a master key which has never been transmitted. When registers or subkeys are flushed out after single use, the hacker or side channel attack has no knowledge of the master key.

After the subkeys or registers generate the initial key stream, two bytes are drawn from that stream, appended together, and pushed through an S-box. Two bytes entered and only one byte emerged which becomes part of the key stream that is actually used. This is a software function and not a hardware function so the side channel attack can provide no information about it. Additionally, there is no information gathered from a side channel attack on subkeys when they are represented by physical registers that can give any information of the bytes that emerge from the S-box.

Side channel attacks require mapping physical data against cipher text. With Whitenoise all operations are order 1 (XOr) so there is no fluctuation or pattern in the cipher text. It is like a flat line on an ECG. The attack cannot correlate the overlap between the physical data and the cipher text. It turns the attack into a brute force game and long before anything can be determined the dynamic one-time-pad key will have changed and the hacker has to keep starting over each time.

Brute force attacks

" In cryptography, a brute-force attack, or exhaustive key search, is a cryptanalytic attack that can, in theory, be used against any encrypted data[1](except for data encrypted in an information-theoretically secure manner." Wiki

The University of Victoria was unsuccessful in this kind of attack over months of trying with a super computer array.

Simple logic defense

All testing with super computer arrays show this is not possible. The keys are exponential and every variable is variable.

Quantum computing attacks

As we prepare for the future a deeper look at the implications of quantum computing attacks is merited.

“The two best known quantum computing attacks are based on Shor's algorithm and Grover's algorithm. Of the two, Shor's offers the greater risk to current security systems.”

“Derivatives of Shor's algorithm are widely conjectured to be effective against all mainstream public-key algorithms including RSA, Diffie-Hellman and elliptic curve cryptography. According to Professor Gilles Brassard, an expert in quantum computing: "The time needed to factor an RSA integer is the same order as the time needed to use that same integer as modulus for a single RSA encryption. In other words, it takes no more time to break RSA on a quantum computer (up to a multiplicative constant) than to use it legitimately on a classical computer." The general consensus is that these public key algorithms are insecure at any key size if sufficiently large quantum computers capable of running Shor's algorithm become available. The implication of this attack is that all data encrypted using current standards based security systems such as the ubiquitous SSL used to protect e-commerce and Internet banking and SSH used to protect access to sensitive computing systems is at risk. Encrypted data protected using public-key algorithms can be archived and may be broken at a later time.”

“Mainstream symmetric ciphers (such as AES) and collision resistant hash functions (such as SHA) are widely conjectured to offer greater security against known quantum computing attacks. They are widely thought most vulnerable to Grover's algorithm. Bennett, Bernstein, Brassard, and Vazirani proved in 1996 that a brute-force key search on a quantum computer cannot be faster than roughly 2n/2 invocations of the underlying cryptographic algorithm, compared with roughly 2n in the classical case.[10] Thus in the presence of large quantum computers an n-bit key can provide at least n/2 bits of security. Quantum brute force is easily defeated by doubling the key length, which has little extra computational cost in ordinary use. This implies that at least a 160-bit symmetric key is required to achieve 80-bit security rating against a quantum computer.” Wiki

Given that the smallest whitenoise key possible is 1600 bits (10 times stronger than the above example) and that scope documents recommend using keys that are orders of magnitude stronger than that quantum computing attacks are not a problem.

Simple logic defense

"Not problems" might be solvable with quantum computing unless there is NO constant variable. In Whitenoise all variables are variable.

Mathematical attacks

Mathematical attacks compromise mathematical functions used in traditional cryptography. The University of California, Berkeley could find no known mathematical attacks that are effective against Whitenoise.

Simple logic defense

Whitenoise keys are created from a mechanical process and not a mathematical process so there is no mathematical function to "put in reverse" and unwind.