Lookout

WHITEPAPER

NotCompatible.C

A sophisticated mobile threat that puts protected networks at risk

Introduction

Malicious actors now view mobile devices as a viable attack vector and the attacks have reached new heights of operational sophistication. In 2012 Lookout first reported on the discovery of NotCompatible, an Android threat disguised as a system update that turned compromised devices into TCP proxies controlled by the attackers. Since that initial discovery, Lookout has tracked the evolution of the cybercrime group responsible and the increased technical sophistication of the latest variant, NotCompatible.C. Given the potential security risk this latest variant poses to enterprise networks, we encourage security organizations to increase monitoring of mobile device network activity and deploy protection against attacks of this kind.

NotCompatible.C contains proxy functionality that allows attackers to infiltrate secure enterprise networks via compromised devices. This whitepaper examines the construction and operation of NotCompatible.C, explores the network security risk, and proposes methods for protection.

This threat features impressive new technical attributes compared to earlier variants, attributes that in combination Lookout has never before observed in a mobile threat. These attributes include:

1. Resiliency

NotCompatible.C is resilient to network-based blocking because it uses a peer-to-peer protocol and has multiple, geographically-distributed Command and Control (C2) servers. The geo-distribution of its C2 servers allows the malware to function even if law enforcement is able to take down individual servers. Peer-to-peer protocols make the malware resilient to IP and DNS based blocking by enabling infected devices to receive commands by proxy via other infected devices.

2. Resistance to Network-Based Detection

NotCompatible.C encrypts all C2 and proxied data traffic end-to-end while also performing mutual authentication between clients and C2 servers via public key cryptography. This protocol-level encryption can prevent network security systems from being able to differentiate malicious traffic from legitimate traffic.

3. Self-Protection

NotCompatible.C uses a Gateway C2 to analyze incoming connections and likely prevents active probing of the various Operational C2s by blocking connections from non-approved IP addresses.

The operators behind NotCompatible.C have built up their population of infected devices on the back of massive spam campaigns and a lack of mobile threat protection on device populations. So far the attackers have not pushed malicious APKs to the Google Play Store and their unsophisticated, but effective distribution methods stands in stark contrast to the sophistication of their backend architecture and design.

Devices that operate outside the traditional security perimeter at most organizations represent a weak point in an otherwise layered security defense. These devices can be compromised using malware such as NotCompatible.C and attackers have begun to capitalize on this opportunity. Hand-held inventory scanners infected with malware, for example, were recently used by attackers to bypass perimeter security defenses and steal a company's entire financial database[1].

NotCompatible.C presents attackers with an opportunity to access protected networks by allowing attackers to access any network a mobile device connected to, including corporate Wi-Fi and VPNs. Lookout has analyzed traffic for NotCompatible.C clients connected to "generic" private networks and has not seen evidence of automatic network scanning; however, we have not yet analyzed traffic from infected devices on potentially targeted corporate networks. NotCompatible.C's observed use to date in this sandboxed environment has revolved largely around sending spam

[1] Marko, Kurt. "How a Scanner Infected Corporate Systems and Stole Data: Beware Trojan Peripherals." Forbes 10 Jul. 2014: http://www.forbes.com/sites/kurtmarko/2014/07/10/trojanhardware-spreads-apts/

and bypassing e-commerce anti-fraud mechanisms as attackers can route a large volume of transactions through a geographically distributed network of devices that appear as legitimate sources for traffic.

In summary, NotCompatible.C stands as an unacceptable backdoor to have on any device connected to an enterprise's internal network. Lookout urges enterprises to implement detection to identify infected devices and enforcement to prevent such devices from connecting to Wi-Fi and VPN.

The Technical Sophistication of NotCompatible.C

Summary

While the "A" variant of NotCompatible was a relatively simple piece of malware, the family has since taken on many of the features found in mature PC malware and has become what is arguably one of the most sophisticated mobile threat operations ever seen.

- NotCompatible.C, the latest variant, features:
- UDP and TCP protocol support
- Peer-to-peer communication between compromised devices
    - A sophisticated two-tiered C2 architecture:
    - A back-end which has been designed to be resilient and secure through the use of multiple, geographically-distributed Operational C2s
- A back-end which resists active probing through the use of a Gateway C2
- End-to-end encryption of all C2 traffic and proxied data
- Mutual authentication of clients and C2s through public key cryptography

Origins

In 2012 when Lookout first detected the NotCompatible family (i.e variant "A"), it acted as a simple proxy on infected devices. NotCompatible.A used a simple client-server architecture where the client communicated directly with one C2 at a time. All communication took place without any encryption or obfuscation to hide the activity, making it trivial for network-based defenses to detect and block.

The observed client traffic proxied by NotCompatible.A was simple, containing mostly HTTP and SMTP traffic. The primary usage of this variant revolved around fraudulent ticket purchases and spam[2].

NotCompatible.A's distribution occurred primarily through "drive-by-download" attacks where victims are automatically served a malicious mobile application when they visit a website. Attackers using drive-by-downloads to spread malware will often attempt to social engineer the user into installing the application. In the case of NotCompatible.A, the operators infected many legitimate websites with drive-by-downloads. NotCompatible's operators also used spam campaigns from hacked email accounts to direct users to drive-by-downloads, an effective technique that resulted in global detections of NotCompatible skyrocketing up to 20,000 detections a day when these spam campaigns were active[3].

When NotCompatible's operators later developed the new "C" variant, they used similar distribution tactics, however with a slower roll-out process. We believe this slow roll-out represented a testing phase for the new, more complex, backend architecture.

Server Architecture & Operations

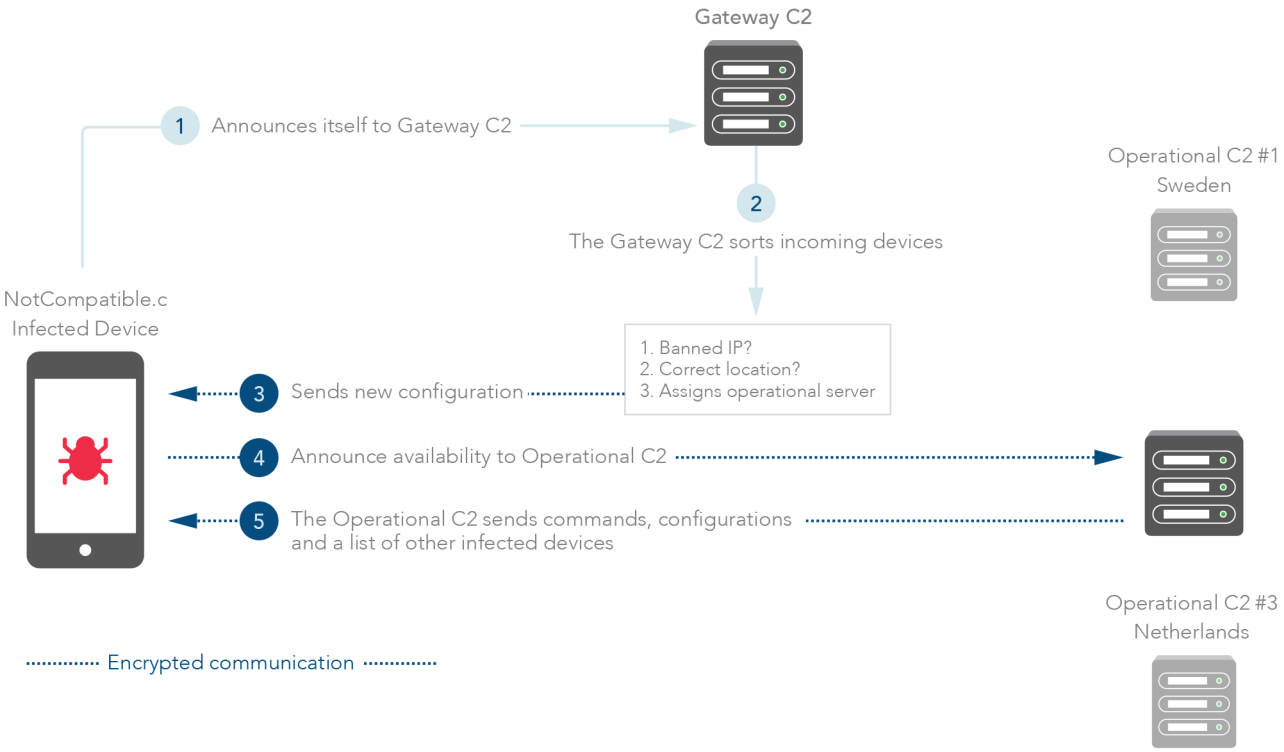
NotCompatible.C uses a two-tier command and control network architecture. The first layer is comprised of a Gateway C2 and the second layer is comprised of Operational C2s.

[2] Strazzere, Tim. "Security Alert: Hacked Websites Serve Suspicious Android Apps (NotCompatible)." 2 May 2012: https://blog.lookout.com/blog/2012/05/02/security-alert-hacked-websites-serve-suspicious-android-apps-noncompatible/

[3] Strazzere, Tim. "Still NotCompatible: A Resurgence Via Email Spam." 14 Mar. 2013: https://blog.lookout.com/blog/2013/03/14/still-notcompatible-a-resurgence-via-email-spam-2/

The Gateway C2s perform load balancing of inbound client traffic across multiple Operational C2s and access control by only allowing authenticated clients to connect to the Operational C2s. Operational C2s are ultimately responsible for controlling the flow of proxied data over infected clients.

NotCompatible.C Operation



Notably, the Gateway C2 would often be unavailable to all clients attempting to make a connection. While this could be the result of bugs or improper configurations, the outages occurred on a regular basis and may have represented an effort to minimize C2 exposure to outside parties.

While observing these outages, we also noticed what appeared to be selective responses by the Gateway C2. Sometimes clients were ignored by the Gateway while others were accepted and then redirected to an Operational C2. The Gateway's decision to ignore or accept a client appeared to be based in part on IP address similarity among a group of clients so the operators may have been segmenting the clients geographically to use the traffic more efficiently.

Another interesting possibility raised by this two-tiered C2 network architecture is that it could make the discovery of the Operational C2s difficult for behavioral analysis systems and researchers. Lookout observed that some IP addresses were always blocked, thereby discouraging interrogation of these C2s by outside parties. If the Gateway C2 filtering mechanism works properly and IP spaces not corresponding to mobile devices were blocked, then a dynamic analysis environment would not pick up on any traffic between the Gateway C2 and the sandboxed environments, which could result in the incorrect conclusion that the malware sample was "dead" or not malicious.

If the Gateway C2 accepts a client, it transmits a configuration file containing all active Operational C2s. At last count, there were more than ten distinct Operational C2s distributed around the world. From our analysis, we believe that clients are directed to connect to a specific Operational C2 according to their geography so that clients in the same region will connect to similar Operational C2s, just as Internet content delivery networks (CDNs) route browsers to the closest server with the content they need. For example, when Lookout researchers attempted to connect to an Operational C2 based in Europe from a device based in Asia, they were redirected by that C2 to another Operational C2 based in Europe. These operational C2's are spread around the world in various countries including Sweden, Poland, Netherlands, United Kingdom and the USA.

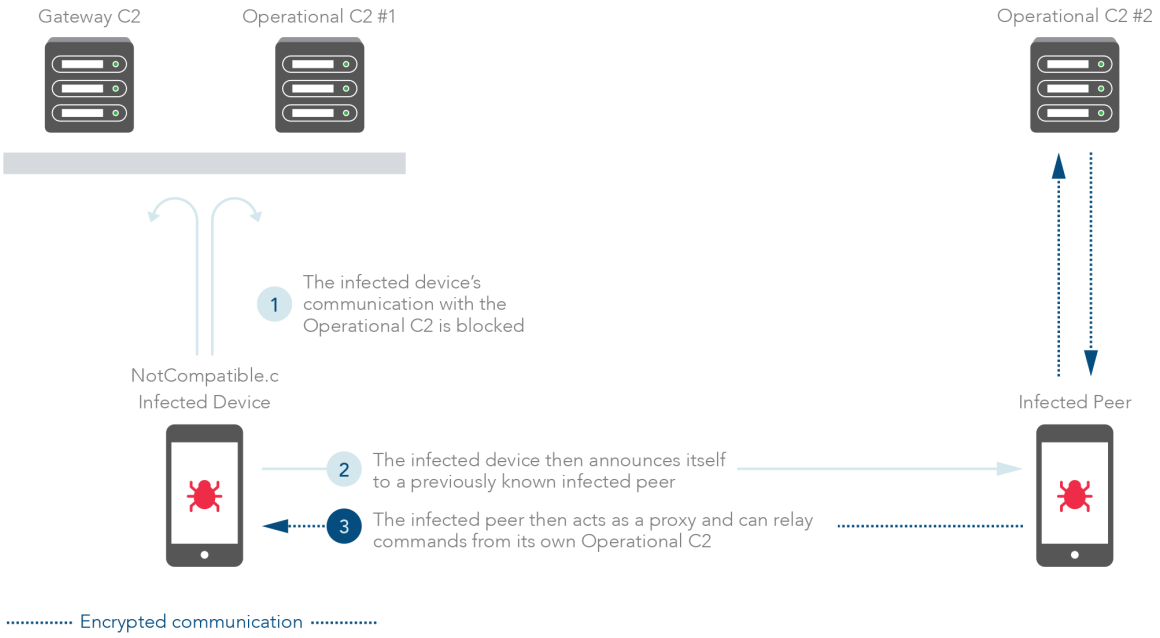
Client Connections

While interacting with an Operational C2, a client may issue a command to retrieve a list of peers to communicate with over TCP or UDP (a HUBLIST or UDPHUBLIST command, respectively, which contain these lists of infected peers.

For a detailed list of commands from both clients and C2s please refer to the Appendix). Upon receiving this command, the Operational C2 transmits a list of other infected clients to which the client can connect. After receiving this list of peers, the infected client will attempt to connect to peer clients while keeping its connection to the Operational C2 open. Upon the first infected client establishing a connection to a second peer client, through either TCP or UDP, the two clients will assess which of the two has a newer configuration file. If one has a newer configuration, it will share it with the other.

The capability for clients to share operational C2 server addresses in a peer-to-peer network creates a powerful redundancy in the NotCompatible.C infrastructure that hardens it against disruption. In theory, a network security solution used by organizations could identify and block an Operational C2; however, if a compromised device can find another device that is not blocked then it could continue to receive C2 commands by proxy, as illustrated below:

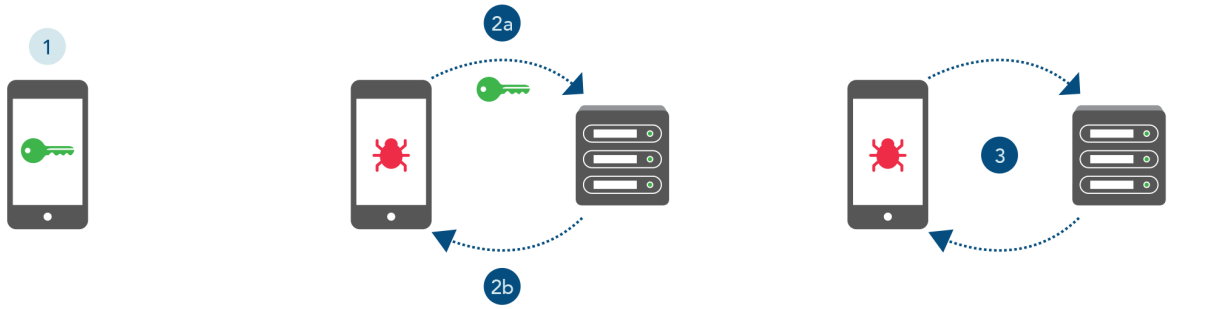
NotCompatible.C Peer-to0Peer Communication



C2 Protocol Structure and Encryption

NotCompatible.C differs from earlier variants with the addition of encryption to the network protocol. The protocol between infected clients and both Gateway and Operational C2s is illustrated below. A similar protocol without a pre-shared public key is used when peer clients communicate with each other.

NotCompatible.C Peer-to Peer Communication

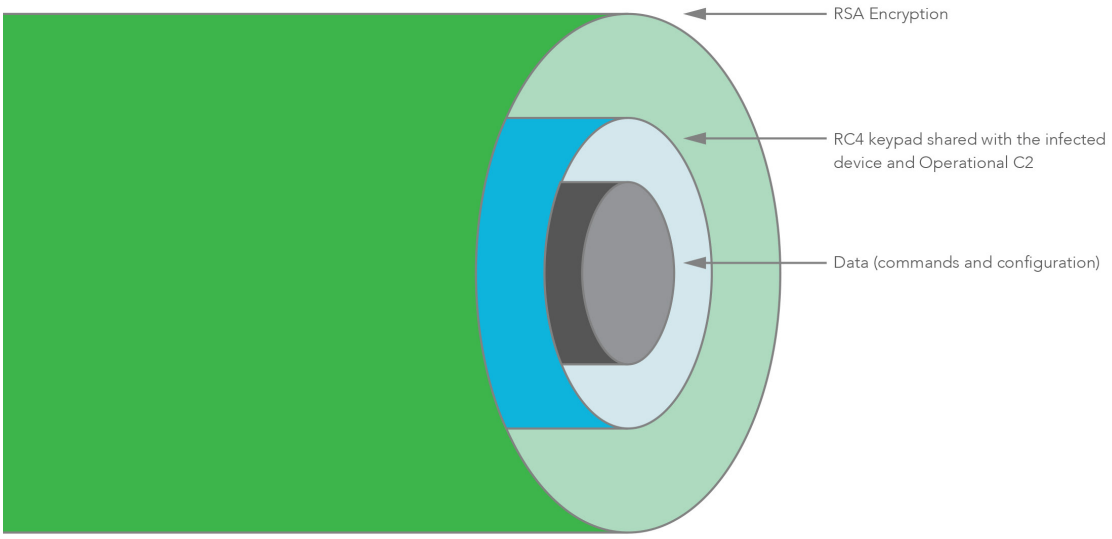


1. The infected client generates an RSA public key.
2. The infected client encrypts its RSA public key with the C2's public key and transmits it to the C2.
3. After decrypting the client's RSA public key, the C2 then initializes an RC4 shared key for use during future communications with the infected client. It then encrypts that RC4 shared key with the client's RSA public key and sends it to the client. The client receives and decrypts the shared RC4 key from the C2.
4. Any subsequent transmission between the client and the C2 will be encrypted first using the shared RC4 key and then the receiver's (client or C2) public key. The C2's public key has remained static in all samples observed by Lookout researchers.

Once this key exchange is finished, no traffic is transmitted in cleartext. Though this protocol is custom to NotCompatible, it blends in with other SSL traffic on a network by using port 443 and transmitting data using seemingly valid SSL records. Notably, there is no SSL session initialization in this protocol, making it stand out from legitimate SSL traffic.

Below shows the layers of encryption between the client and the C2 once the key exchange has completed:

NotCompatible.C Encrypted Traffic



Distribution Methods

NotCompatible.C appears to be distributed in the same manner as earlier variants, using drive-by-downloads through spam campaigns and compromised websites. It appears that the operators behind the campaigns have bought compromised accounts and websites in bulk since the campaigns can vary quite significantly. Lookout researchers observed NotCompatible spam campaigns where each campaign used a different block of compromised accounts. For example, one campaign used compromised accounts from AOL, while another used compromised Yahoo! accounts.

NotCompatible.C does not use any exploits to install, preferring social engineering tactics to trick victims into completing installation. For example, one spam email we have observed targeting Korean companies informs the user that they need to install a "security patch" in order to view an attached file. Other spam emails advertised weight loss solutions and some included nothing more than a link that served an APK to Android devices.

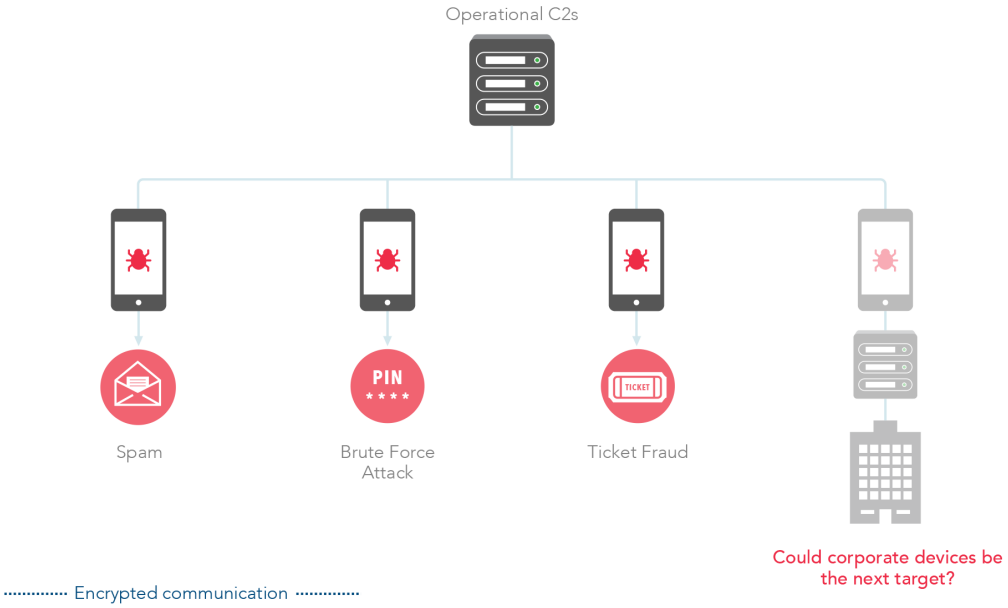
Current Attack Targets

By monitoring the traffic proxied through NotCompatible.C, we have observed a wide variety of malicious traffic, leading us to believe that either NotCompatible.C's operators are a large, multi-faceted cybercrime group or are providing access to their network to other cybercrime groups. Some example forms of traffic we have observed include:

- Spam campaigns—Compromised Live, AOL, Yahoo!, and Comcast accounts used to send spam (e.g. weight loss advertisements).
- Bulk ticket purchasing—Ticketmaster, Live Nation, StubHub, and Craigslist bulk ticket purchases, bypassing IP reputation anti-fraud mechanisms in place on these sites.
- Bruteforce attacks—Password guessing for Wordpress blog administrator portals. This activity was low volume with little observable pattern, and may have been an attempt to expand the infection vector of NotCompatible via compromised sites.
- c99 shell control—Logging into PHP-based backdoors dropped onto vulnerable sites that are used to carry out malicious activities (e.g. defacing the site).

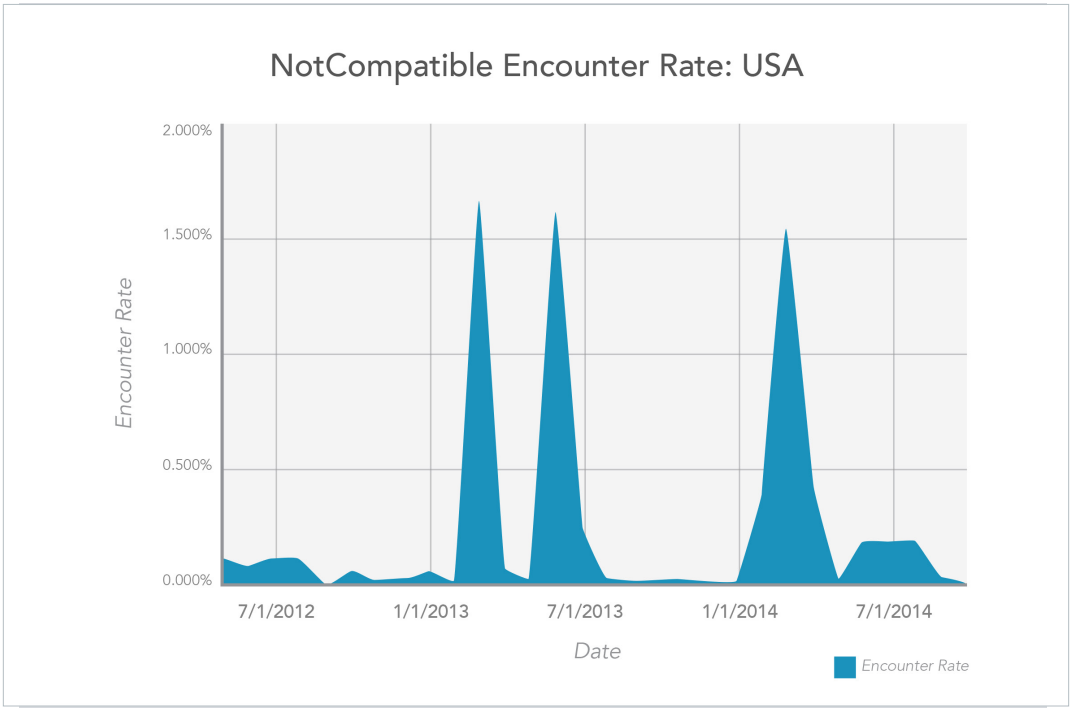
NotCompatible.C and the Risk to Protected Networks

NotCompatible.C Attack Patterns

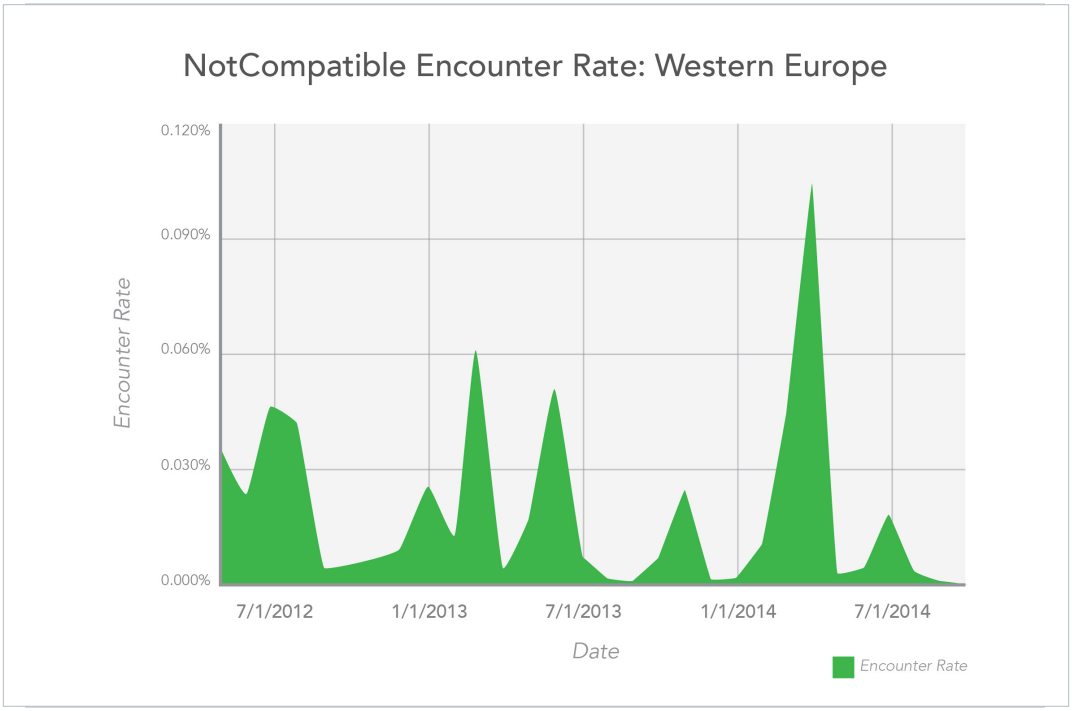


If the operators of NotCompatible.C are in fact providing access to their network to other cybercrime groups the possibility that it could be used to attack corporate networks cannot be ignored, especially given the prevalence of NotCompatible, which has reached mobile device encounter rates of over 1% in the United States at the height of its distribution:

Graph 1



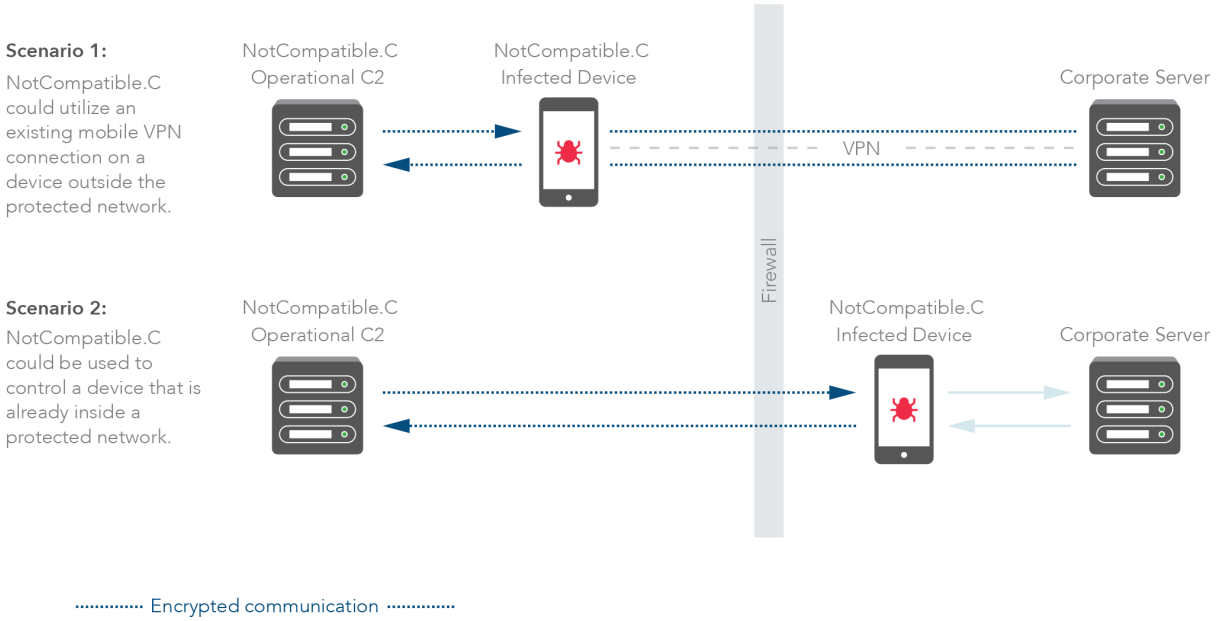
Graph 2



We believe that NotCompatible is already present on many corporate networks because we have observed, via Lookout's userbase, hundreds of corporate networks with devices that have encountered NotCompatible. It's reasonable to assume there are many more devices with active NotCompatible infections that are not protected by Lookout that also connect to corporate networks.

At its heart, NotCompatible.C is an unrestricted proxy on a mobile device that offers the operators unfettered access to protected networks to which these devices connect.

NotCompatible.C an Enterprise Risk?



Because infected devices connect to a C2 infrastructure, attackers may target a particular organization by analyzing which network a device connects from. An infected client present on an enterprise network would potentially allow attackers to enumerate vulnerable hosts inside the network, exploit vulnerabilities in these hosts and exfiltrate data.

NotCompatible.C is particularly difficult for network-based security systems to detect or block. First, traffic appears to typical network security systems as SSL data on port 443 (the default SSL port). Further, the designers of NotCompatible.C have used existing cryptography libraries sensibly, leaving no obvious flaws that could be used to detect or decrypt its traffic. Peer-to-peer communication allows infected clients to remain connected (via their peers) to C2 infrastructure even if enterprises or network operators block communications with known C2 hosts at the network layer.

Lookout has analyzed traffic on infected devices connected to "generic" private networks and has not seen evidence of automatic local network scanning; however, we have not yet analyzed traffic from infected devices on potentially targeted corporate networks. We urge enterprises to step up monitoring for mobile devices performing any sort of local network scanning, which would be indicative of a targeted threat such as NotCompatible.C, and implement protection strategies to prevent such devices from connecting to Wi-Fi and VPN.

Protection Strategies

Implement mobile threat protection: Mobile devices typically operate outside the traditional perimeter and beyond the reach of network-based security solutions. An advanced mobile security platform allows organizations to monitor for and protect against suspicious activity on their mobile devices, block identified threats and assess the overall health of their mobile ecosystem. Next generation threats such as NotCompatible.C can provide access to protected networks and facilitate the exfiltration of data in a way that most enterprises are not prepared to defend against. By detecting this threat at the device level, it is possible to block and prevent installation before an attacker can perform any hostile activity.

Segment the network: All mobile devices used in protected environments—especially those able to connect to external unmanaged networks—should be limited to an isolated network segment with strong controls limiting access to sensitive resources and analytics to detect potentially malicious behavior.

Conclusion

NotCompatible.C possesses unique and impressive technical sophistication in the world of mobile malware. Its resiliency, resistance to network-based detection, and self-protection features make it a potent threat in the hands of an attacker. As a mobile botnet with widespread distribution and proxy capabilities, the potential use of NotCompatible.C as a gateway to attack protected networks and systems is not only plausible, but a likely outcome.

Apendix

C2-Commands

I. Client-to-Client P2P Commands	
Helo	Initialization packet
GETSET	Send settings back to client
GETHUB	Send Hub-List's (p2p and udp) back to client
HUBLIST	Receiving p2p Hub-List-Data
UDPHUBLIST	Receiving udp Hub-List-Data
SET	Receiving a settings packet
SUIP	Receiving data of what this clients current ip and port is (for further propagation)
LID	Settings-LID (allowing clients to know who has a more current one)

II. Client-to-Client UDP-Commands	
SUIP	Receiving data of what this clients current ip and port is (for further propagation)
PING	Send PONG (Heartbeat)
PONG	Send PING (Heartbeat)
PINGME	Send PINGTO command to all hosts on UDP list for the client asking for a ping
PINGTO	Ask a client to ping a different client (propagation od clients)
SET	Receiving settings packet
GETSET	Send settings packet back to clients
LID	Settings-LID (allowing clients to know who has a more current one)
GETHUB	Send-List's (p2p and udp) back to Client
UDHUBLIST	Receiving p2p Hub-List-Data
GETPOINT	Receiving udp Hub-List-Data
PUSH	"Push" to a UDP point (IP-Address that is not a client)
PUSHTO	Initialize PUSH to an UDP point
LIST	Receive a listof Endpoints to talk with

Apendix

C2-Commands

III. Client-to-C2-Commands	
PING	Send PONG (Heartbeat)
HUBLIST	Receiving p2p Hub-List-Data
UDPHUBLIST	Receiving udp Hub-List-Data
SETGROUP	Sets with server group from the settings packet shouldbe used for further connections
SET	Receivin settings packet
SETV	Use an indexgroup inside of the server group
CONN	Open a connection proxy to a specific host
SHUT	Close connection proxy
SEND	Send (and receive reply) with data through Proxy