chronicle

WHITE PAPER

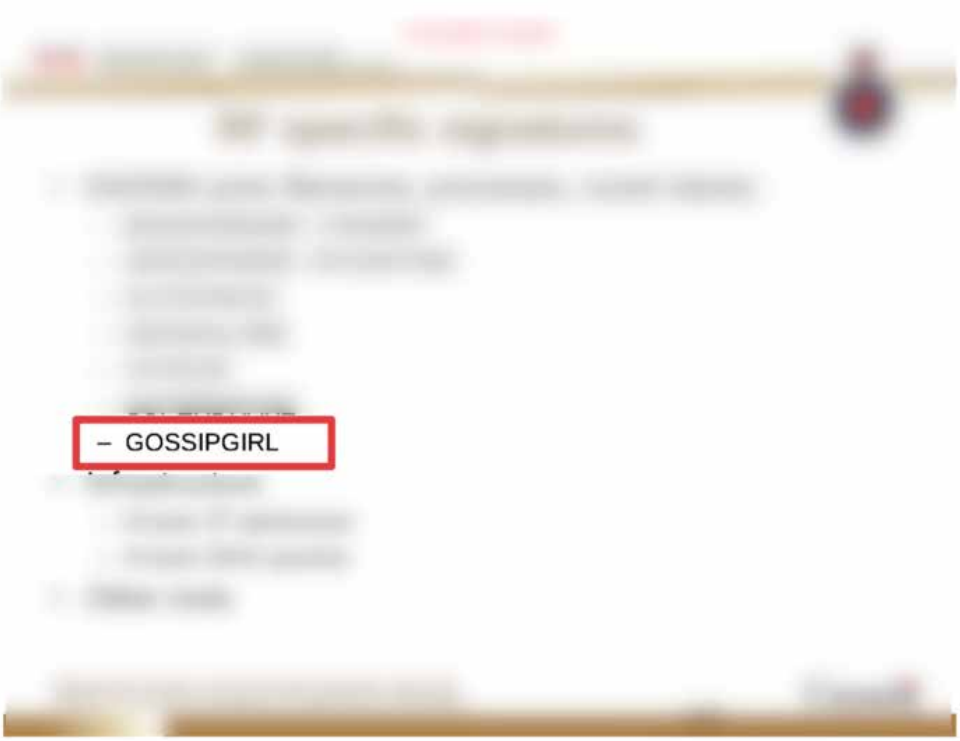
Who is GOSSIPGIRL?

Revisiting the O.G. Threat Actor Supergroup

Threat intelligence as a discipline is the continuous pursuit of cyber situational awareness. The idea that there are threat actors out there, unbeknownst to us, operating in the shadows, with impunity is something that drives researchers to hunt further and further. The more unsettling experience is having a small piece of information on a threat without context: a single indicator, a rumor, …a name. That maddening uncertainty drove us from a vague name on a creative revisiting of a cluster of the most prolific and daunting threat actors ever discovered. And along the way, we were given the opportunity to fill in important gaps in the history of humankind's incursion into the fifth domain.

Studying Our Favorite Slidedeck

Of the multitude of documents leaked from sensitive operations over the past six years, few are as rich and interesting as a CSEC presentation titled "Pay attention to that man behind the curtain: Discovering aliens on CNE infrastructure". Bullet listings meant to serve as a presenter backdrop are often too anemic to provide meaningful food for thought in the absence of the speaker, but that's not the case here. The CSEC slidedeck discusses different methods for conducting Counter-CNE (i.e.: threat intelligence). Among the fascinating insights is a list of threat actor cryptonyms including names like SEEDSPHERE, ALOOFNESS, and VOYEUR.

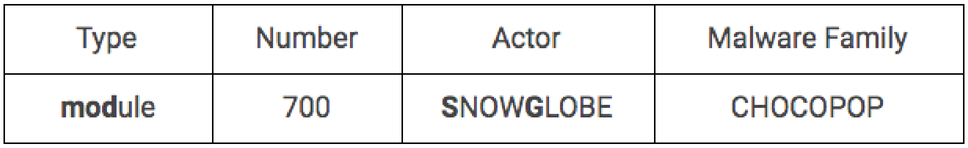


– GOSSIPGIRL

Slide highlighting some of the threat actors tracked in REPLICANTFARM by their cryptonyms

Some of these actor names, like MAKERSMARK and SNOWGLOBE are now familiar to us. The former now a common name related to Moonlight Maze, Agent.BTZ, and Turla. The latter (as the presentation goes on to detail) is another name for Animal Farm and includes its well-known Babar malware family. But who are these other actors worthy of rising to the scrutiny of our northern siblings?

We combed through the presentation inch by inch, hoping some small fragment would slip out that we'd be able to associate to the unidentified actors. A small sliver of hope presented itself in Slide 23. A pixelated screenshot of an alerts window shows barely legible signature names like mod_101_MM_CARBON — a likely reference to **M**AKERS**M**ARK's (i.e.Turla's) Carbon system. That would mean the format of these signatures is:



Type	Number	Actor	Malware Family
module	700	**S**NOW**G**LOBE	CHOCOPOP

Combing through the other legible names, we were largely stumped. The actor acronyms were either unrecognizable or the malware family names were unfamiliar or of little hunting value. However, one signature stood out, 'mod_501_**GR**_FLAME.pl'. Was this a connection between **G**OSSIPGI**R**L and Flame? The thread was thin but it was all we needed to set us off on our parallel investigation.

GOSSIPGIRL on a Blank Slate

Flame (a.k.a Flamer or sKyWIper) was the object of extensive research and fascination by the security community circa 2011. Crysys Lab, Kaspersky Lab, and Symantec researchers contributed extensive analyses on this modular cyberespionage platform. While some in the security community were skeptical of the hype generated by claims of yet another modular platform discovered soon after Stuxnet and Duqu, Flame impressed all with the discovery that it employed a novel cryptographic attack to impersonate a Windows Update server within an enterprise and spread onto other computers as if legitimately signed by Microsoft itself. Ultimately, it turned out that the fortuitous discoveries of Stuxnet, Duqu, and Flame were in fact related beyond superficial succession.

So how best to start an investigation into what CSEC might've been looking at with Flame? Well, we could turn to a different leak. As part of the Shadow Brokers leaks, the security community got access to a series of malware signatures for a program identified as Territorial Dispute ('TeDi'). The TeDi signatures are an interesting (if likely outdated) snapshot of Five Eyes Counter-CNE efforts and, as such, a great starting point for rebuilding what might be categorized under the GOSSIPGIRL moniker. Dr. Boldizsár Bencsáth published excellent work identifying a portion of the TeDi signatures. Among them, two signatures for Flame (SIG9 and SIG16) as well as a signature for its likely predecessor, Miniflame (SIG10).

Applying Modern Tooling to Ancient Problems

It's important to remember just how nascent threat intelligence really is as a field. Pre-2010, the anti-malware industry considered cyberespionage a possibility, or perhaps an eventuality, but not an ongoing phenomenon. What followed was a dizzying period of discovery that made the industry realize that malware-based digital espionage was not only going on, but had been going strong, unnoticed for years. The tranquil days of reverse engineering banking trojans were pierced by Stuxnet, Duqu, Flame, Gauss, and MiniFlame. Each a complex modular malware family worthy of its own pedigree, most abusing zero-day exploits, and used to carry out operations imbued with geopolitical gravitas.

However, the rapid rate of discovery often leads modern practitioners to mistakenly assume that the tooling available to the researchers involved in these discoveries is any way comparable to our own. Remember that at the time YARA rules were not widely used for malware research. Nor did researchers have access to VirusTotal retrohunts, code similarity searching at scale, ngram or byte-string searches, etc. Most of the modern tooling that makes hunting for malware scalable was not available. Instead, these researchers were relying on tooling proprietary to each AV company and stitching their visibility together as best they could to form a picture. The limitation only makes their discovery more admirable but it also made us wonder 'what did they miss?'

Ultimately, we decided to take creative license with the GOSSIPGIRL term and use it for two purposes: to compensate for an ontological deficiency in threat intel terminology, and, to investigate a collaborative umbrella of threat actors.

Describing Supra Threat Actors

Private sector Threat intelligence has largely shied away from abstract methodological discussions. Our concepts have largely served us well, except when it comes to outliers like mercenary APTs or fourth-party collection practices. However, ignoring methodological deficiencies leads to problematic blind spots. In particular, the one-to-one equivalence of a 'threat actor' to an institution or organization has left us incapable of accurately representing multi-institution, multi-country, or multi-group orchestration in collaborative operational deployment, platform development, or generally complex deconfliction practices.

The resulting limitations are best exemplified by Regin as a likely multi-country platform, Equation as a likely multi-institution umbrella, and Red October as a possible dual-country collaboration. While each of these 'threat actors' was researched extensively and competently, the industry was ill-equipped to describe the perpetrator arrangement faithfully. For that, we want to introduce the concept of a Supra Threat Actor or STA.

The introduction of a supra category of activity clustering isn't meant to add complexity to an already jargon-filled space but rather to allow us to faithfully describe the interesting phenomenon of multi-threat actor or multi-platform operations. The staple of these (as opposed to a threat actor that uses two or more closed-source malware families) is a formalization of cross-platform compatibility that allows droppers, payloads, and configuration files be co-opted and instrumented by the closed-source tooling of other threat actors. We have seen multiple examples of this next-level development practice amongst the true apex threat actors.

A notable example is appears to be the case of the Wzowski API that allows the CSEC and DSD's WARRIORPRIDE, GCHQ's DAREDEVIL, and NSA's STRAITBIZZARRE and UNITEDRAKE tooling to work together despite independent development practices.

With GOSSIPGIRL we'll be discussing another example of a collaborative threat actor umbrella.

All Roads Lead to Stuxnet

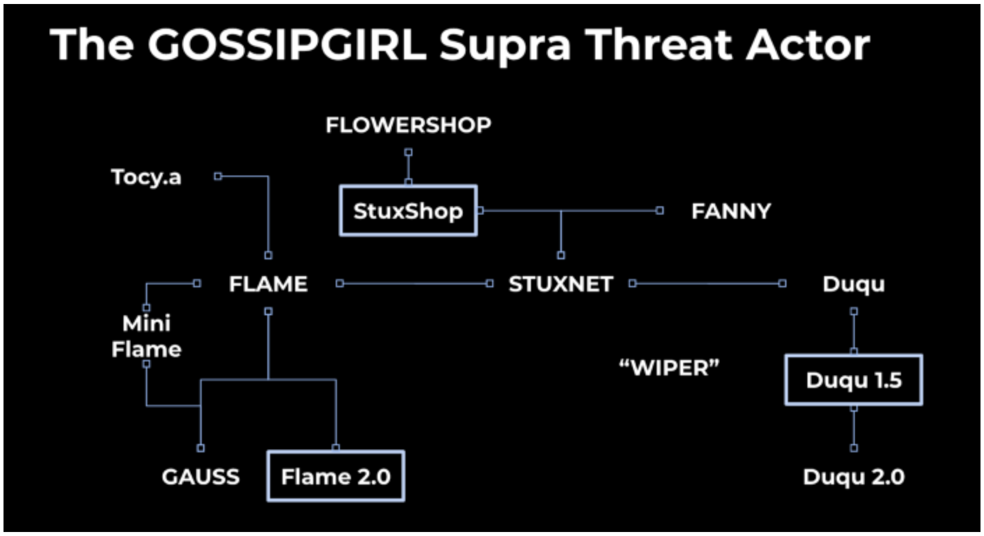
As we conducted our parallel investigation into Flame we found ourselves retracing steps from one malware platform to another, much like the researchers that originally discovered them. First from Flame to Mini-Flame, it's likely predecessor, and then to Gauss, considered a more widespread successor or perhaps a side operation. However, despite the lore of Gauss (due to its never-cracked encrypted payload), we didn't find a live lead there. We were impressed by the relative simplicity of Mini-Flame and the completely restructured modular architecture of Gauss, but ultimately it seemed that Flame had died with the deployment of the SUICIDE module in May 2012.

Instead, we shifted our focus onto Stuxnet. In June 2012, Kaspersky researchers discovered that the older version of Stuxnet included a Flame plugin, Resource 207. This connected Flame as one of the threat actors involved in the development of Stuxnet. And it ultimately placed Stuxnet in a central role for our research into the GOSSIPGIRL STA. If we are going to treat GOSSIPGIRL as a collaborate supergroup of threat actors, then Stuxnet represents the fruit of their collaboration. As such, Stuxnet placed at least two other threat actors within the scope of our research: Duqu and Equation.

Researchers connected Duqu to the development of Stuxnet early on. Main Stuxnet kernel drivers (like 'mrxcls.sys') shared developmental links with Duqu's 'Tilde-D platform', involving the threat actor in some of the central development of Stuxnet. Equation, on the other hand, would eventually be connected by the use of exploits shared by both Stuxnet and an earlier Equation Group worm named Fanny. Fanny utilized two Stuxnet zero-days 1–2 years before Stuxnet entered the scene: the infamous LNK exploit (CVE-2010–2568) and a privilege escalation embedded in the aforementioned Resource 207. Furthermore, Kaspersky researchers would note shared coding practices between the Stuxnet and Equation developers.

Choosing to stand on the shoulders of the research giants that discovered these operations, we decided to dive once again into the malware. We hoped that leveraging technology and insights unavailable to our predecessors would yield a greater understanding of the GOSSIPGIRL STA. The resulting discoveries were beyond our wildest expectations.

Our Discoveries in a Snapshot

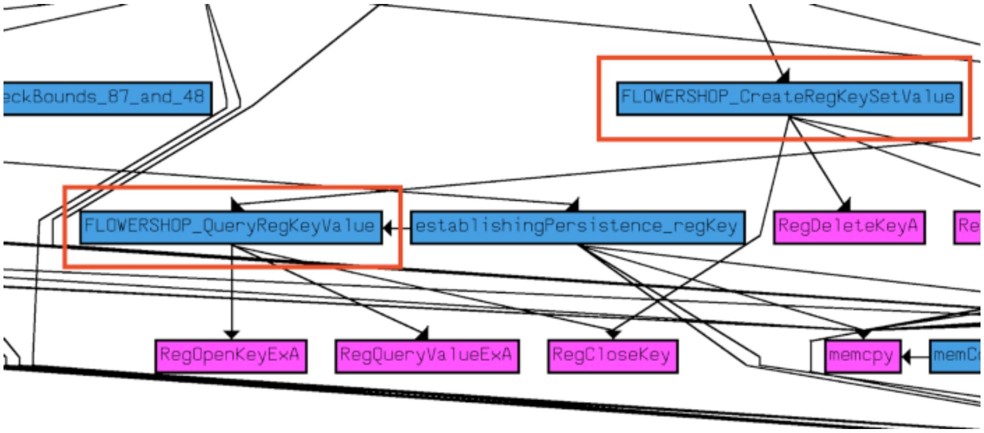


GOSSIPGIRL Supra Threat Actor cluster of interrelated actors and malware platforms including new discoveries.

After casting such a wide net for insights on what was supposed to be old news, we actually fished out more discoveries than we could handle. Each of these merits a technical deepdive in its own right and links are provided for a technical writeup detailing each. We hope other researchers, malware analysts, and defenders will benefit from the breakdowns and technical indicators. In brief:

STUXSHOP: The Oldest Stuxnet Component Dials Up

Stuxshop is ancient component folded into Stuxnet to manage it's early command-and-control capabilities. The discovery of Stuxshop is significant precisely because it unveils the presence of a fourth team involved in the early development of Stuxnet. Stuxshop shares unique code overlaps with Flowershop (a.k.a. TeDi: SIG17/SIG18, Cheshire Cat), a malware platform active from 2002–2013 with targets across the Middle East.


eckBounds_87_and_48
FLOWERSHOP_CreateRegKeySetValue
FLOWERSHOP_QueryRegKeyValue
establishingPersistence_regKey
RegDeleteKeyA
Re
RegOpenKeyExA
RegQueryValueExA
RegCloseKey
memcpy
memC

Stuxshop function call graph highlighting identical embedded Flowershop functionality

Key takeaways:

- Stuxshop ties a fourth threat actor into the GOSSIPGIRL Supra Threat Actor responsible for Stuxnet.
- Stuxshop further exemplifies the modular design that produced Stuxnet, as a ‘plane flown as it’s being built’.
- It lends credence Symantec’s hypothesis that Stuxnet was in development as early as 2005, as the Stuxshop compilation timestamp signals 2006 and the relevant overlapping Flowershop codebase was seen in-the-wild as early as 2007.

Read Stuxshop technical analysis (pdf): https://storage.googleapis.com/chronicle-research/STUXSHOP%20Stuxnet%20Dials%20In%20.pdf

Duqu 1.5: A Ghost in the Wires of a Diplomatic Venue

An investigation into a ‘Ghost in the Wires’ infection at a venue for diplomatic talks revealed a missing link in the development from the Duqu 1.0 threat actor involved in Stuxnet to the formidable Duqu 2.0 in-memory modular platform discovered in the Kaspersky offices and P5+1 talk venues in Switzerland. This story highlights what can be accomplished when an excellent in-house incident response team collaborates with threat researchers to hunt down an elusive threat.

Duqu 1.5’s experimental modular design

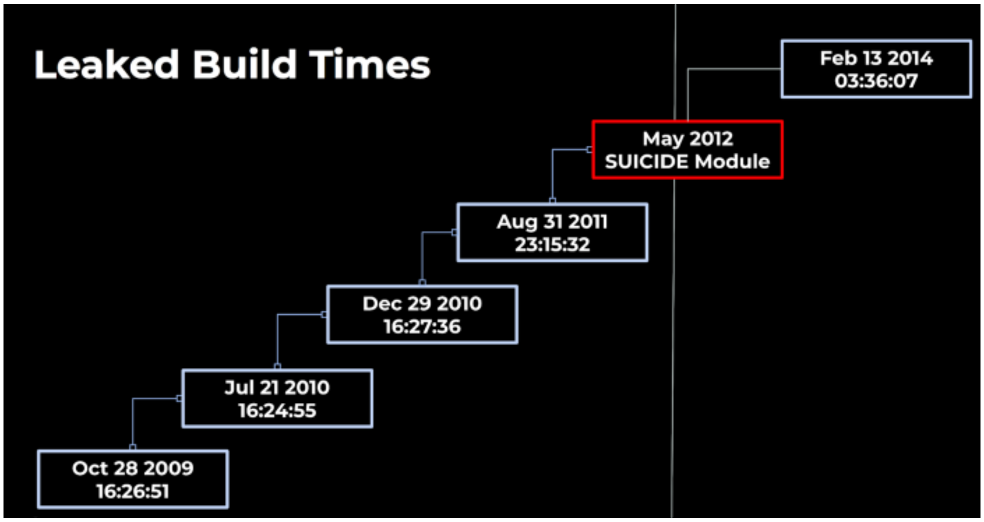
Key takeaways:

- An incident response team’s tenacity for answers revealed artefacts to recreate a full Duqu 1.5 infection across the network of a diplomatic venue.
- The missing link reveals the iterative development efforts that lead from the burned Duqu 1.0 version of the ‘Tilde-D’ platform to the fully in-memory implementation of Duqu2.0.
- It includes a more bloated, experimental, multi-tier loading chain: it starts with a trojanized floppy kernel driver signed with a stolen certificate to load up a registry virtual file system (VFS), that loads an in-memory orchestrator, which then loads an on-disk VFS, to deploy a series of plugins for further spreading and backdoor access to infected machines.
- If it weren’t for the watchful stance and reflex of the incident response team, Duqu 1.5’s multi-tier structure would’ve kept researchers from ever rebuilding all the components of this infection. Moreover, two different AV companies analyzed the first-stage kernel driver without flagging it as malicious. At the time of writing, static detection ratio remains at zero.

Read Duqu technical analysis (pdf): https://storage.googleapis.com/chronicle-research/DuQu%201.5%20A%20Ghost%20in%20the%20Wires.pdf

FLAME 2.0: Risen from the Ashes

Finally, the third and perhaps most momentous finding was our discovery that Flame merely faked its own death. In May 2012, the Flame operators issued a SUICIDE module to clean up active infections and burned down their remaining command-and-control infrastructure. While this move successfully drove researchers away from tracking Flame, in reality they were simply retooling. At this time, we are releasing technical indicators and initial findings of a new iteration, Flame 2.0, and inviting the research community to collaborate with us in investigating its resurgence.



Mapping leaked build times show's a development beyond deployment of Flame 1.0's SUICIDE module

Key takeaways:

- Some Flame 2.0 components continued to leak the original build time of an underlying statically-linked PuTTY library whenever the attackers forgot to remove debug symbols.
- Despite timestomping the sample compile times to look older, the leaked build times show that Flame 2.0 samples were compiled as early as February 2014, nearly two years after the deployment of the SUICIDE module.
- Flame 2.0 includes the first Flame samples compiled for 64-bit Windows systems.
- The attackers now employ AES-256 to encrypt the second-stage embedded resources. At this time, we've been unable to crack the encryption to reveal what new scripts and payloads are deployed by Flame 2.0 after a successful infection.

Read Flame 2.0 technical analysis (pdf): https://storage.googleapis.com/chronicle-research/Flame%202.0%20Risen%20from%20the%20Ashes.pdf

Parting Insights

We are often asked why we bother investigating older threat actors. Why bother with 'cyber-paleontology'? The reasons are manifold:

Firstly, research into past incidents allows us to leverage new tools and insights that weren't available at the time of initial discovery. At the time Stuxnet, Flame 1.0, and Duqu 1.0 were discovered, YARA rules weren't widely used, VirusTotal's retrohunt capability weren't available, nor did we have access to scalable code similarity engines.

Secondly, retrospective research allow us to unearth connections between diverse threat actors and malware platforms that expand our understanding of the notable adversaries in cyberspace and their respective institutional configurations. These insights speak to humankind's incursion into the fifth domain. If we fail to study and document these incidents while the source data is still available, it may prove impossible in the future.

Finally, if it's not evident by now, retrospective research yields unexpected surprises. While the research community assumed Flame had retired and ceased to track this ominous threat actor, Flame 2.0 samples appeared in VirusTotal as early as October 2016 and were likely available in private AV collections a year or two before that. Given that Flame proved to be one of the most daring threat actors ever discovered (going so far as to leverage an innovate MD5 hash collision attack to subvert the Windows Update mechanism to spread infections across an enterprise), this isn't an adversary we should take lightly in our remit to defend the internet ecosystem.

From a methodological standpoint, we hope that the research community will take cautious advantage of a higher ontological category to describe collaborative frameworks for multiple threat actors. GOSSIPGIRL isn't the first supra threat actor (STAs) unearthed by the research community, it's only the first to be described in comprehensive terms. A focus on this 'multi-tenant' model of modular malware development and deployment should allow for a higher-fidelity understanding of: the trends followed by seemingly diverse threat actors, the closed-door sharing of techniques and tools, and the organizational complexities behind clusters of malicious activity that defy simplistic attribution claims.

Discovering, understanding, and continuously monitoring complex attacks is difficult and requires a mixture of specialized talent, tooling, and visibility that most enterprises and institutions don't have access to. It's unrealistic to expect every company to develop these capabilities in-house. However, as the Duqu 1.5 example shows, the collaboration between driven incident responders and threat researchers yields insights that benefit our shared defense mission in a way neither could accomplish on their own.

A better understanding of the institutions and incentives involved in cyberespionage further supports the view that threat actors don't go away after exposure; our aggressors never truly vanish. They have an intelligence remit to fulfill and will go to great lengths in doing so. The defender community must be willing to match these efforts in order to insure the collective safety of users and organizations that lack the resources to defend themselves against the most formidable threat actors.

One got outed…
…one went to sleep…
…one faded into memory…
…and one simply played dead.