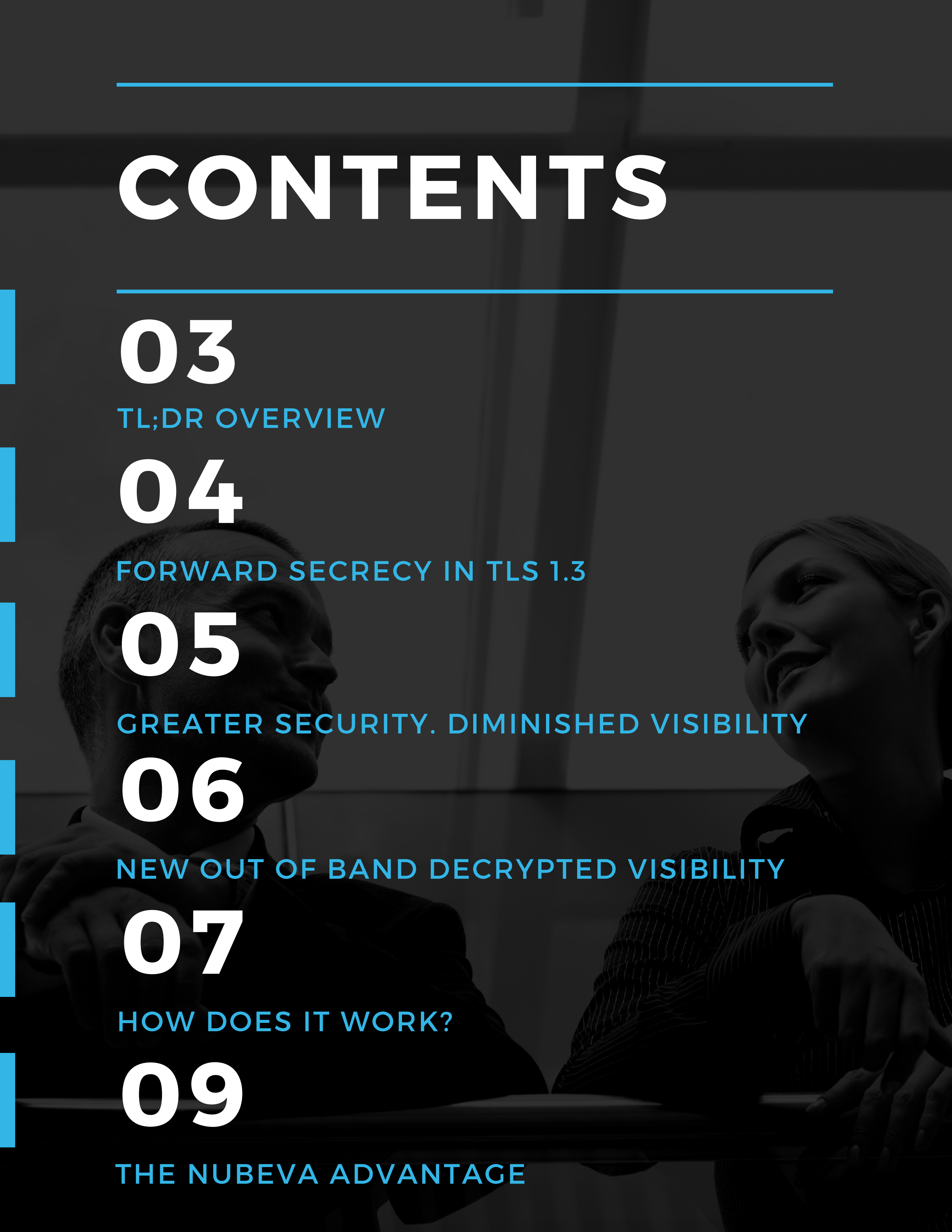
NUBEVA

OUT OF BAND DECRYPTION IN TLS 1.3

TECHNICAL EBOOK

JULY 2019



CONTENTS

OVERVIEW

MODERN TLS DECRYPTION: NOT JUST FOR THE NEXT-GENERATION BUT FOR THE CLOUD GENERATION.

Forward Secrecy in TLS 1.3 makes network communications more secure but also renders traditional out-of-band, man-in-the-middle and decryption at cloud-scale untenable.

The intent of the new TLS 1.3 standard is that, if you want to inspect and monitor traffic, you must do so at the endpoints because everything else is locked down with new, stronger ciphers, rapidly rotating "ephemeral" keys and certificate encryption.

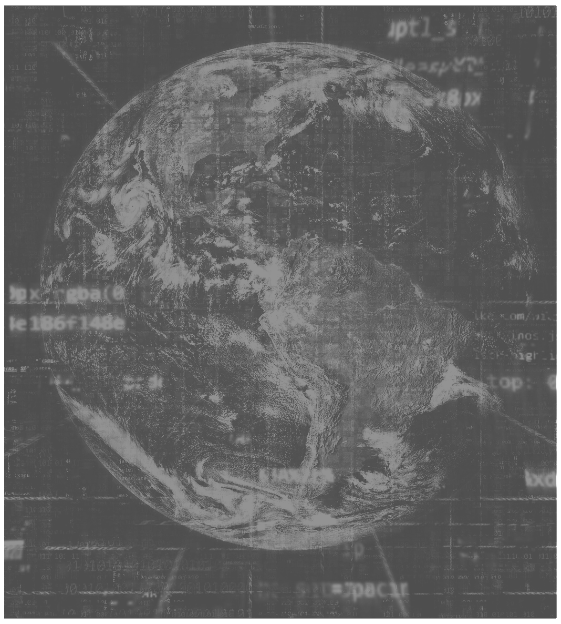
Nubeva has created a new, modern architecture that not only makes out-of-band decryption possible for TLS 1.3, it does so in a way that works with any cloud platform, any packet capture or brokering service and without requiring code changes, architectural changes or workload impact.

This paper explains the challenges posed by forward secrecy in TLS 1.3, the reasons traditional MITM and out of band solutions don't work and how Nubeva's new, architectural approach is superior.

Nubeva is the choice not just for the next-generation, but for the cloud generation.

/ FORWARD SECRECY IN TLS 1.3 /

FORWARD SECRECY IS A DECADES-OLD CONCEPT THAT IS NOW REQUIRED IN TLS 1.3.



Forward secrecy means 2 things.

First, it means that each session is separately encrypted. This makes it much harder for bad actors to compromise data in motion. Even if one session is captured, subsequent sessions will still remain safely encrypted.

Second, it means that both the TLS client and the TLS server participate in the encryption key creation for each session. The TLS handshake starts like before. Then, using separate processes, the client and the server use the asymmetric key to arrive at a final, “symmetric” key. Because this final key is used only for the duration of the session, it is ephemeral. This final, symmetric, ephemeral key is what actually encrypts and decrypts the packet traffic for each individual session.

These ephemeral keys can also be renegotiated periodically for additional security (e.g. renew every minute). Over the course of a day a pair of workloads communicating back and forth may generate thousands of separate, final, ephemeral keys, even though the public/private keys and certificates remain the same. If a server has 1000 sessions it could be generating 1000 unique, final keys per minute.

As mentioned before, TLS 1.3 requires forward secrecy. This all means that data in motion, especially data in the cloud and travelling between workloads of distributed (i.e. containerized / microservice architected) applications, is much better protected than before.. That protection comes at the cost of visibility.

/ GREATER SECURITY. DIMINISHED VISIBILITY. /

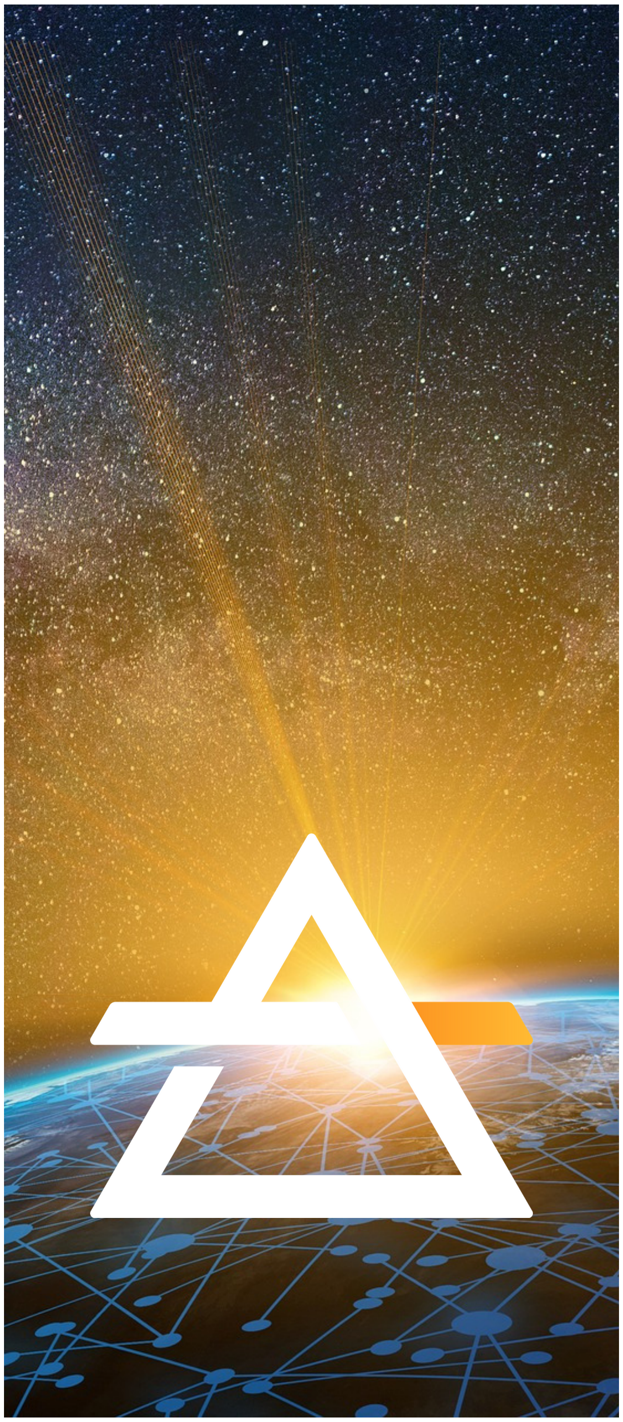
TLS 1.3 poses a challenge to cybersecurity tools and processes. Network security tools like intrusion detection systems, behavioral analysis systems, data loss prevention systems, threat hunting and forensic systems and others inspect packets as they traverse the network. These systems look for malware and other indicators of compromise or bad actors. These tools must inspect decrypted packets. Where before they could rely on a single pre-shared key or certificate information to decrypt every message, with TLS 1.3 they cannot. Indeed, they are hindered even in TLS 1.2 settings where PFS is used. There are thousands of ephemeral keys now and certificate information is now, itself, encrypted.

Inline tools like firewalls and intrusion prevention systems will be set up as a man-in-the-middle architecture to act as a proxy for incoming traffic. They will participate in the TLS handshake as if they were the intended client and decrypt, inspect, then re-encrypt the traffic and send it on. This MITM method becomes quickly untenable in highly distributed environments where you cannot afford to put expensive in-line firewalls in between each and every workload that is sending packets back and forth. Furthermore, with forward secrecy, any MITM proxy set up now has to handle thousands of TLS handshakes where before they had to handle just one. Even if deployed, this becomes prohibitively expensive and tremendously slow from a performance standpoint.

Because only the endpoints of TLS 1.3 sessions have access to the final, ephemeral encryption keys, security appliances that are out of band or deployed in a MITM architecture that are ***not*** also proxies are blind to the packet traffic payloads they need to inspect. This creates a conundrum. Out of band architectures for security and monitoring tools are preferred since they do not disrupt network communications, impact application performance or induce latency. But they are severely impacted by the new requirements of TLS 1.3.

From a security perspective, TLS 1.3's new PFS requirement and updated cipher requirements is great for external privacy. There are no more "keys to the kingdom" for decryption. And even if you get your hands on some ephemeral keys you need the exact segments of packets for that part of a session, in exact order for it to work. It’s good security, and carries much lower risk on the keys than before. But it comes at the cost of visibility and the resultant ability to inspect, secure and control your data and your applications with your tools, your processes and your teams.

/ NEW OUT-OF-BAND DECRYPTED VISIBILITY. /



Nubeva did something different. Nubeva looked at the problem and realized a few things.

- **First**, out-of-band is far preferable to in-line / MITM for inspection, detection, forensics and monitoring. This is especially true as cloud scale environments and decentralized / distributed application architectures with microservices and containers take off.

- **Second**, getting packets is not a problem. The dataplane (taps / mirrors / brokers) is fine.

- **Third**, decryption is not the problem. If the final keys are known, decryption is fast and efficient.

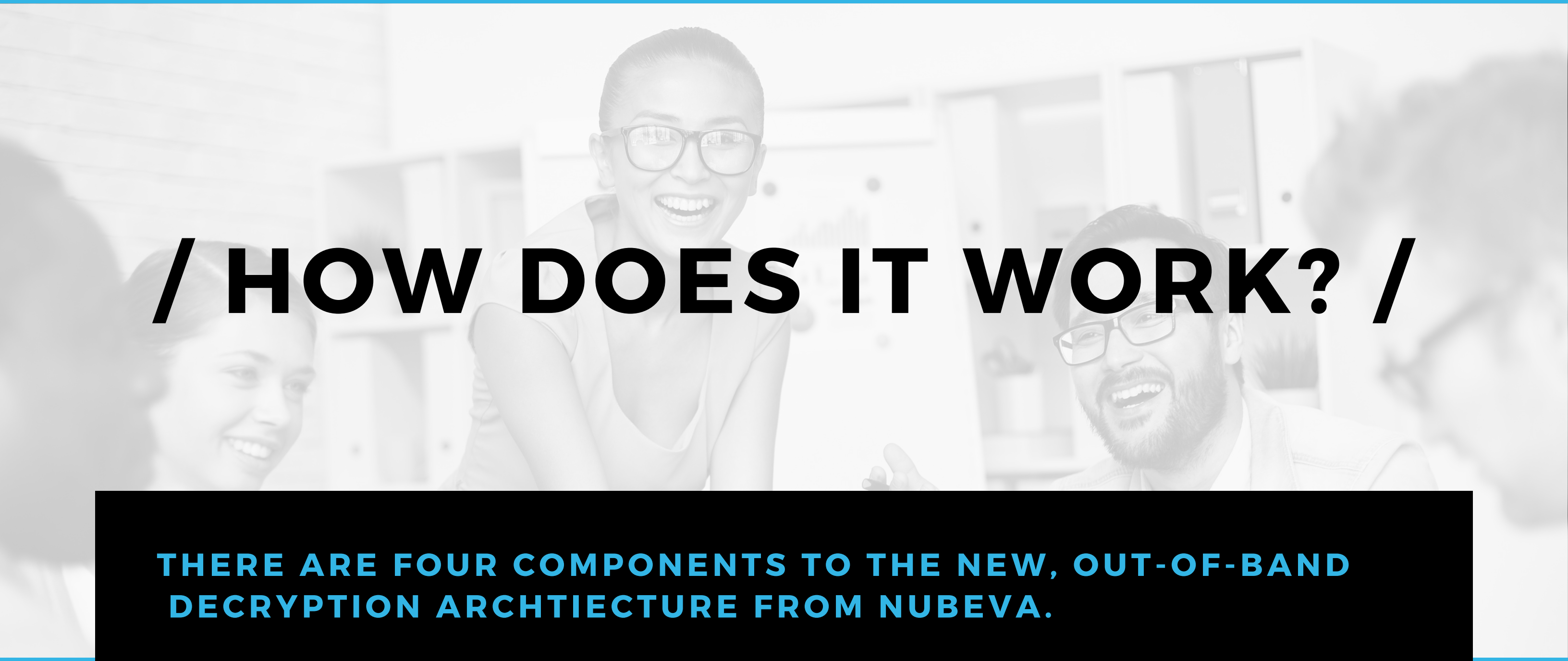
- **Eureka!** the problem is the final, ephemeral keys!

Loading master / private keys into a decryptor is dead in TLS 1.3. So Nubeva figured out how to restore decrypted visibility to existing tool environments. This breathes new life and value into established workflows, tools and teams. It avoids decommissioning out-of-band infrastructure and gives full visibility for even the newest decentralized and distributed cloud environments.

Nubeva Prisms TLS creates an architecture supported by software to restore out-of-band decrypted visibility to any environment:

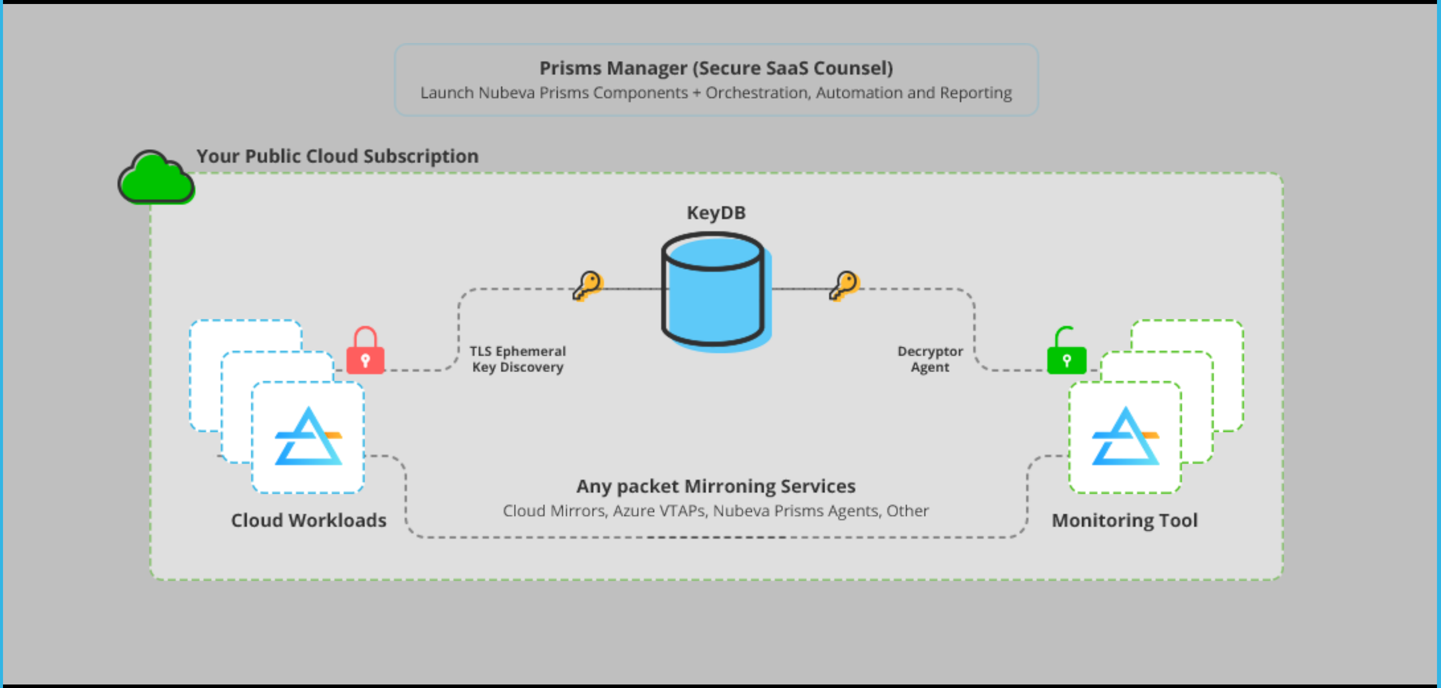
1. Ephemeral / final key discovery and extraction
2. Final key storage in a central key database owned by the customer.
3. Software decryptors that are built to work with ephemeral keys and pull from the customer's central key database.

NUBEVA PRISMS TLS IS AN ARCHITECTURE THAT ESSENTIALLY CREATES A NEW KEY & DECRYPT PLANE SEPARATE FROM THE DATAPLANE.



/ HOW DOES IT WORK? /

THERE ARE FOUR COMPONENTS TO THE NEW, OUT-OF-BAND DECRYPTION ARCHTIECTURE FROM NUBEVA.


Prisms Manager (Secure SaaS Counsel)
Launch Nubeva Prisms Components + Orchestration, Automation and Reporting
Your Public Cloud Subscription
KeyDB
TLS Ephemeral Key Discovery
Decryptor Agent
Cloud Workloads
Any packet Mirroning Services
Cloud Mirrors, Azure VTAPs, Nubeva Prisms Agents, Other
Monitoring Tool

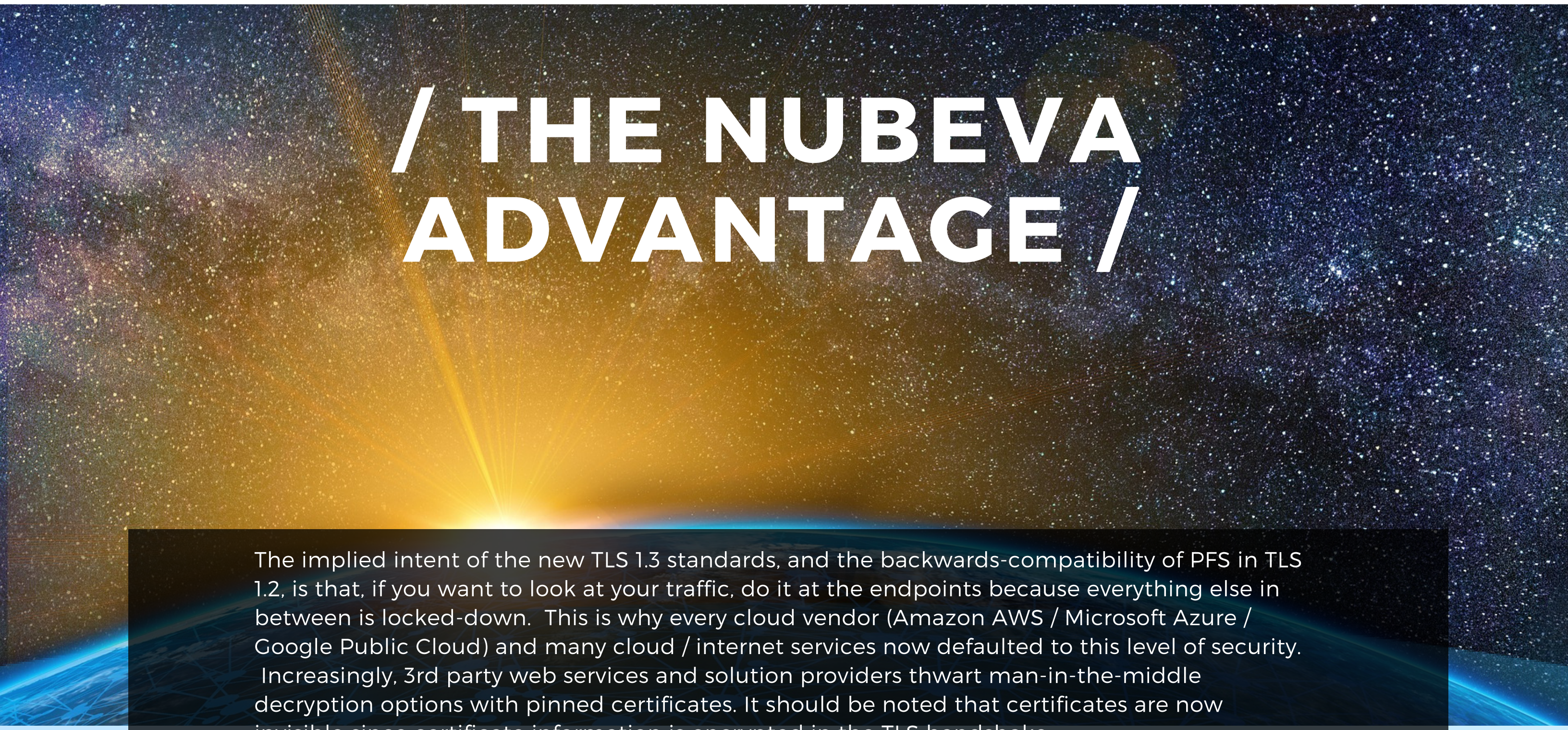
HOW NUBEVA BRINGS BACK OUT OF BAND DECRYPTION FOR CLOUD-GENERATION COMPUTE ENVIRONMENTS.

1 **Final key discovery** relies on **Key Agents**. These are very light, read-only agents that can learn keys from core memory of a host, including native applications and containers running on that host in real-time. Key agents are able to discover final session keys without any application modifications, without library mods, without shims and without debug mode. Behind the scenes, it has cache-based extraction signatures from the management system. The Nubeva management system uses AI-based rules to know where to find the keys for various libraries, system setups, configurations etc. It uses these signatures to very efficiently find the keys are they are created. Because it is discovering and extracting the final keys, an agent is only needed on one-side of the TLS connection; either the TLS client or the TLS server. Agents cannot crash a system because they're insulated from the system workload. Agents can extract over 1 Million keys a minute with <1% CPU and network load. Once keys are identified and extracted, they are sent to the key database.

2 **The final key database** is a centralized cloud platform-based database (e.g. Amazon's Dynamo DB) running in customer's own account, with the customer's own IAM rules. The key database is global, low cost, resilient, and extremely high performance approach supporting billions of keys/day. It is the customer's asset to use anytime / anywhere. Because final keys are securely stored separately from the packet traffic they en/decrypt, decryption is effectively made into a highly scalable and on-demand activity. Encrypted pcaps that were stored for compliance purposes can be retrieved and reviewed at any time. Real-time encrypted streams may be replicated to multiple destinations which may all request and retrieve the relevant final key from the key database at the same time. The keys and encrypted traffic are separated which creates a highly scalable decryption opportunity for any system that needs it.

3 Decryption is handled by soft decryptors that Nubeva has created. These are container-based decryptors which sit on the tool destination environment. When you send the tool encrypted traffic the decryptor detects it and queries the key database, The soft decryptor pulls the final keys tied to that flow and decrypts the packet traffic. The decryptor outputs the newly decrypted traffic as a standard (port 80) packet and it preserves the original TLS packet for complete header telemetry. The decryptors have been tested at > 2Gpbs each on a modest CPU/memory (more throughput than most tools can ingest at).

4 Nubeva TLS decryption is supported by a simple backend SaaS management console to configure the agents and decryptors. However, keys and data never leave the customer's environment. This is critical. Only the management plan is handled outside. As a result, customers get the benefit of SaaS (ease of deployment, operations, scale, cost) but the benefits of in-house security of the data and keys. We call it a split-SaaS model.



/ THE NUBEVA ADVANTAGE /

The implied intent of the new TLS 1.3 standards, and the backwards-compatibility of PFS in TLS 1.2, is that, if you want to look at your traffic, do it at the endpoints because everything else in between is locked-down. This is why every cloud vendor (Amazon AWS / Microsoft Azure / Google Public Cloud) and many cloud / internet services now defaulted to this level of security. Increasingly, 3rd party web services and solution providers thwart man-in-the-middle decryption options with pinned certificates. It should be noted that certificates are now invisible since certificate information is encrypted in the TLS handshake.

The decryption architecture developed by Nubeva delivers amazing features and advantages:

- Nubeva's TLS decryption architecture supports all forms, protocols, ciphers of TLS/SSL.
- It is extensible to future changes due to signature rule based key extraction model.
- It works for native host apps, containers, client and server side sessions.
- Nubeva TLS decryption works with any packet brokering/packet source system that outputs VXLAN packets to the decryptor.
- Nubeva works with any cloud or computing environment.

The Nubeva key agent:

- is available as a container, native linux or Windows Schannel versions.
- It is completely scalable from 1 to 100,000+ monitored sources.
- Is capable of Tbps-capable throughput.

The Nubeva decryptor:

- Works with any tool or monitoring system.
- Requires no application or code changes.
- There are no VPNs to setup or manage for transport of clear text.
- The extracted final keys can work with any other system that can read your keys and your packet captures (e.g. Wireshark).

NUBEVA PRISMS TLS PRESENTS A NEW ARCHITECTURE & APPROACH TO GO WITH NEW ENCRYPTION, NEW APPLICATION ARCHITECTURES AND THE NEW SCALE OF COMPUTING.