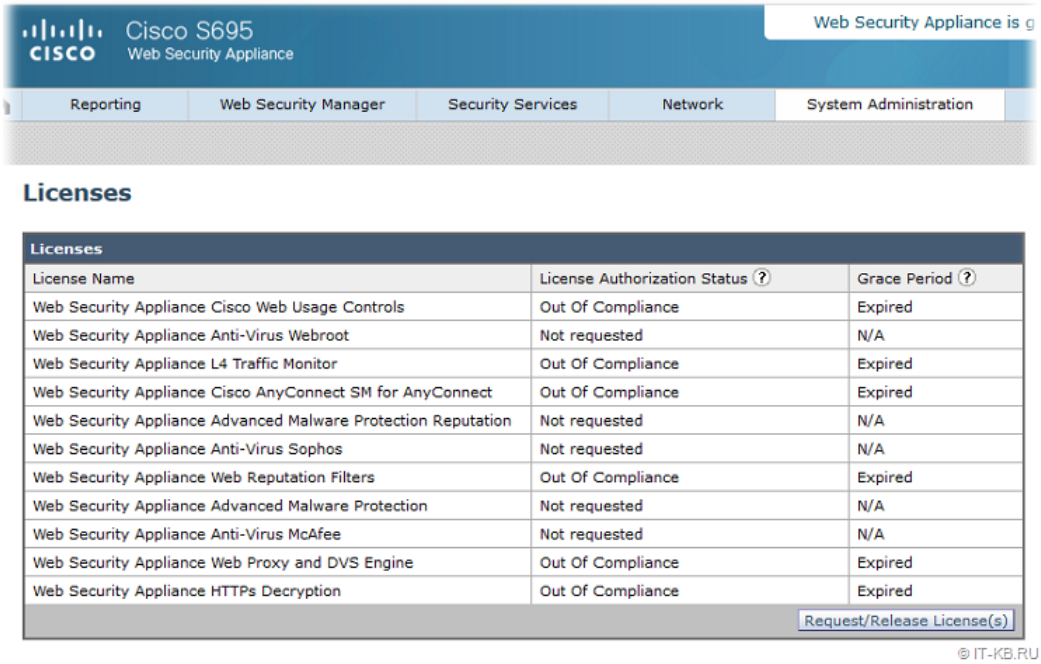
Второе дыхание для серверной платформы Cisco WSA S695 (Cisco UCS C240 M5 Server)

06.09.2022 719 Просмотров

Пару лет назад в одной динамично развивающейся компании было приобретено некоторое количество серверов **Cisco Web Security Appliance** (**WSA**) **S695** под задачу построения управляемой инфраструктуры прокси серверов. Под мою опеку попали два таких сервера довольно не слабой конфигурации: 2 процессора Intel Xeon Gold 6126, 64GB ОЗУ PC4-21300, контроллер RAID SAS 12G с 4GB кеша, расширенная корзина с 16 дисками HDD SAS 12G 600GB 10K, дублированное питание/вентиляторы горячей замены и прочие прелести в корпусе RM 2U.

На борту этих серверов была предустановлена среда управления на базе специализированной ОС Cisco AsyncOS. Лицензировалось всё это дело по временно-действующей подписке и функции прокси работали только либо по активированной действующей лицензии, либо в рамках краткосрочного Grace Period. И вот наступил 2022 год … и Cisco, как и ряд других "закадычных друзей", громко хлопнула дверью, прекратив отгрузки, поддержку и активацию лицензий.

Действующие ранее лицензии на наших серверах истекли, прошёл Grace Period и функции прокси престали работать, превратив серверы WSA в "чемодан без ручки".



Не беда, подумали мы… железо ещё пока модное и вполне бодрое, поставим на него другую ОС и будем использовать под альтернативные задачи.

Но как выяснилось, граждане из Cisco неплохо постарались, чтобы не допустить reuse-сценарий для этого серверного оборудования, рассказывая у себя форуме недоумевающим клиентам, оказавшимся в похожей ситуации, сказки про "вы не понимаете … это ради вашей же безопасности".

Было обнаружено две ключевые проблемы:

- Отсутствовал доступ к контроллеру удалённого правления серверной платформой **Baseboard Management Controller** (**BMC**).
  Он был искусственно заблокирован Cisco и, как я понимаю, загружаемая **AsyncOS** строго контролировала эту блокировку.
- Отсутствовала возможность загрузить сервер с любого накопителя, отличного от загрузочного тома с AsyncOS на RAID-массиве, который был собран Cisco и имел специальную цифровую подпись в загрузчике.
  Блокировка была реализована путём форсированного включения **Secure Boot** в микрокоде **BIOS** с доверием лишь одному загрузочному носителю.

В рамках этой статьи мы рассмотрим вариант решения двух этих проблем.

Общий план действий

Для того, чтобы можно было загружать и устанавливать на сервер ОС отличную от AsyncOS, как то Linux / Windows Server / VMWare ESX, нам потребуется добиться отключения **Secure Boot** для **UEFI**. А если мы захотим поставить на сервер что-то древнее без поддержки UEFI, то может вообще потребоваться переключиться из режима UEFI в режим **Legacy**. В текущих версиях BIOS возможность изменения режима загрузки заблокирована и для того, чтобы вернуть себе эту возможность, нам потребуется понизить версию микрокода BIOS до границы, когда опции управления загрузкой были ещё доступны.

В свою очередь, для того, чтобы без лишних телодвижений управлять версиями BIOS, нам для начала потребуется получить полноценный доступ к контроллеру BMC (в нашем случае это **Cisco Integrated Management Controller** или **CIMC**). Управление контроллером CIMC мы сможем перехватить после того, как на материнской плате замкнём специальный джампер и не дадим больше загружаться **AsyncOS**, а затем отловим интерфейс IMC через **DHCP**.

Прежде, чем приступать, мы должны осознавать возможные последствия и принять некоторые риски:

- Базовая ОС сервера Cisco WSA (AsyncOS) будет удалена и прежняя функциональность WSA доступна уже не будет.
- Ошибки, допущенные в ходе перепрошивок микрокода могут привести в неработоспособное состояние сервер. Однако, даже в этом случае есть некоторые процедуры восстановления микрокода и они описаны в документации.

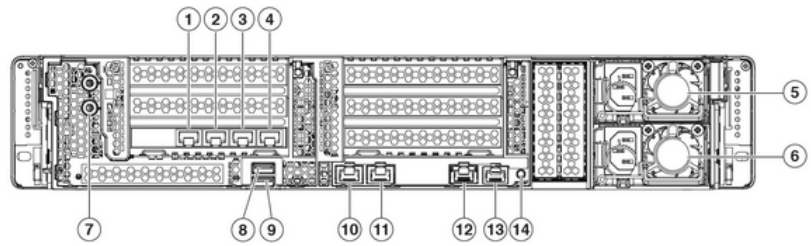
Обратите так же внимание на то, что перед началом операций с заменой микрокода следует обязательно обеспечить серверу бесперебойное электропитание.

Получение доступа к CIMC

В официальной документации на WSA S695 мы не найдём никаких упоминаний или явных инструкций по настройке контроллера IMC, за исключением краткого описания настройки порта с маркировкой "**RPC**": "Cisco Web Security Appliance S195, S395, S695, and S695F Hardware Installation Guide" - "Maintenance and Upgrades" - "Enable RPC" (https://www.cisco.com/c/en/us/td/docs/security/content_security/x95_series/hw/guide/wsa/install-wsa-x95/maintenance.html#id_84088).

Посмотрим на схему описания портов в разделе "Overview" (https://www.cisco.com/c/en/us/td/docs/security/content_security/x95_series/hw/guide/wsa/install-wsa-x95/overview.html):

Figure 11. S695 Rear Panel

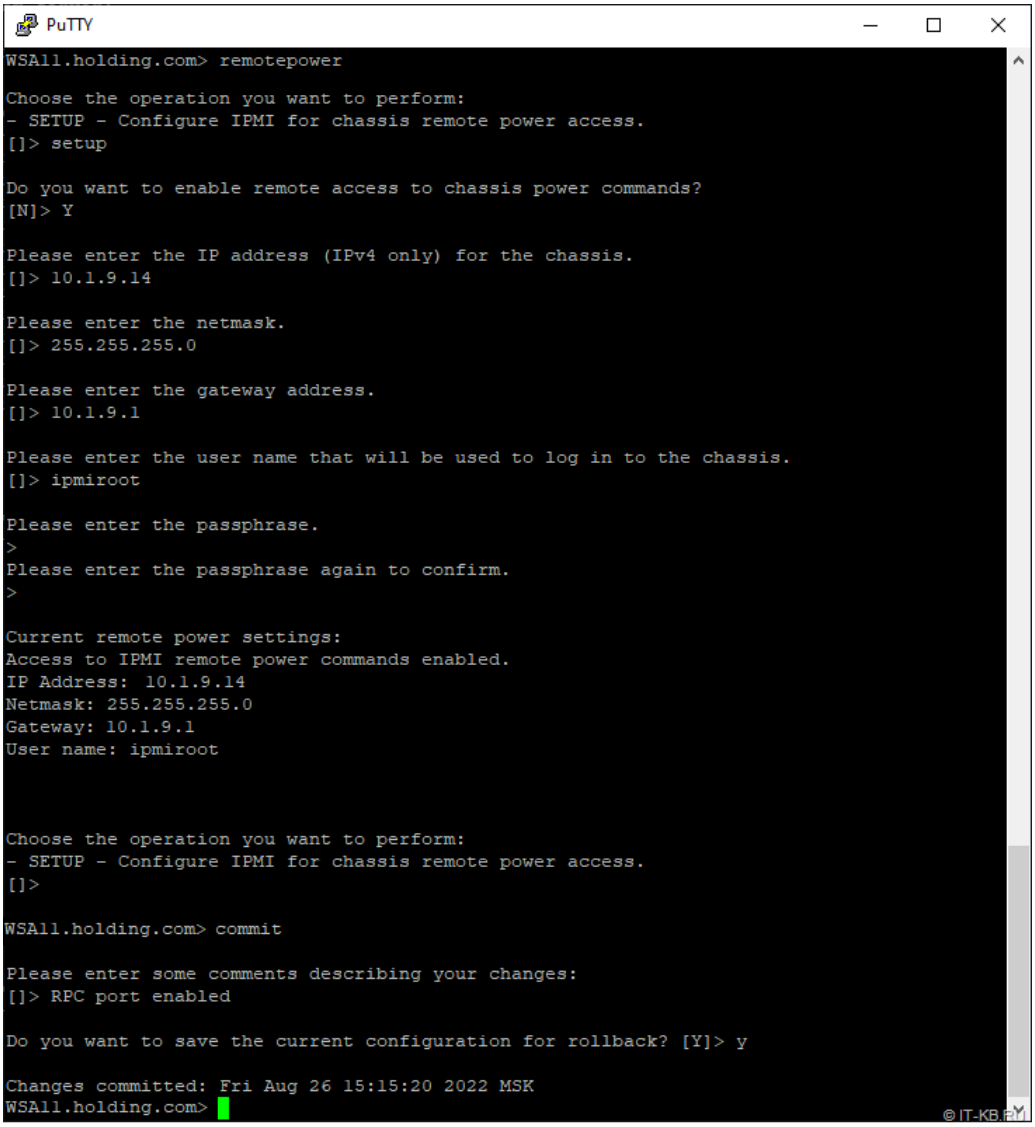


Порты 1-4 используются для работы прокси и мониторинга трафика в предустановленной ОС. Эти порты нам не интересны. На нижней части платформы видим 4 интегрированных в мат.плату порта 10-13

10 - Management interface 1 (MGMT 1). Restricted to management use only
11 - Management interface 2 (MGMT 2). Not in use
12 - RPC port (RPC). Use for remote power cycling.
13 - Serial console port (Console). RJ-45 connector that directly connects a management computer to the appliance.

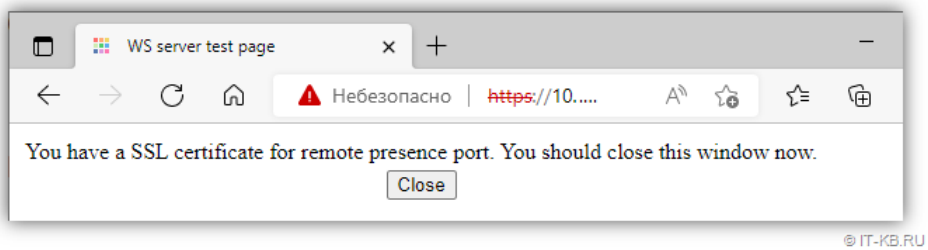
Как мы понимаем, интересующий нас порт "RPC", позиционируется в WSA как порт для удалённого управления питанием через Intelligent Platform Management Interface (IPMI) version 2.0.

Описанная процедура предполагает выполнение команды **remotepower** и её подкоманды **setup**, в ходе которых можно включить и настроить или выключить порт "RPC". В ходе включения и настройки запрашивается IP адрес и учётные данные для доступа к этому адресу. Приведу наглядный пример такой настройки:



После выполнения команды **commit** может пройти 5-10 минут до момента когда порт "RPC" заработает и указанный нами IP станет доступен.

По сути своей эта процедура и есть ни что иное, как первичная настройка контроллера IMC. Однако опыты показали, что WSA намертво отшибает в IMC все протоколы полноценного управления типа SSH или HTTPS. То есть, если пощупать настроенный в AsyncOS IP-адрес интерфейса "RPC", то мы не обнаружим там ничего, кроме единственного открытого порта TCP 2068. Как я понял, этот порт используется в работе vKVM, но как он может нам помочь, мне было решительно непонятно. Описанный в документации пример (https://www.cisco.com/c/en/us/td/docs/security/content_security/x95_series/hw/guide/wsa/install-wsa-x95/maintenance.html#id_84088) с использованием утилиты ipmitool у меня не заработал. При попытке обратиться на этот порт через браузер, я получил невразумительное "You have a SSL certificate for remote presence port. You should close this window now".



Стало очевидно, что по хорошему WSA нам контроллер IMC не отдаст.

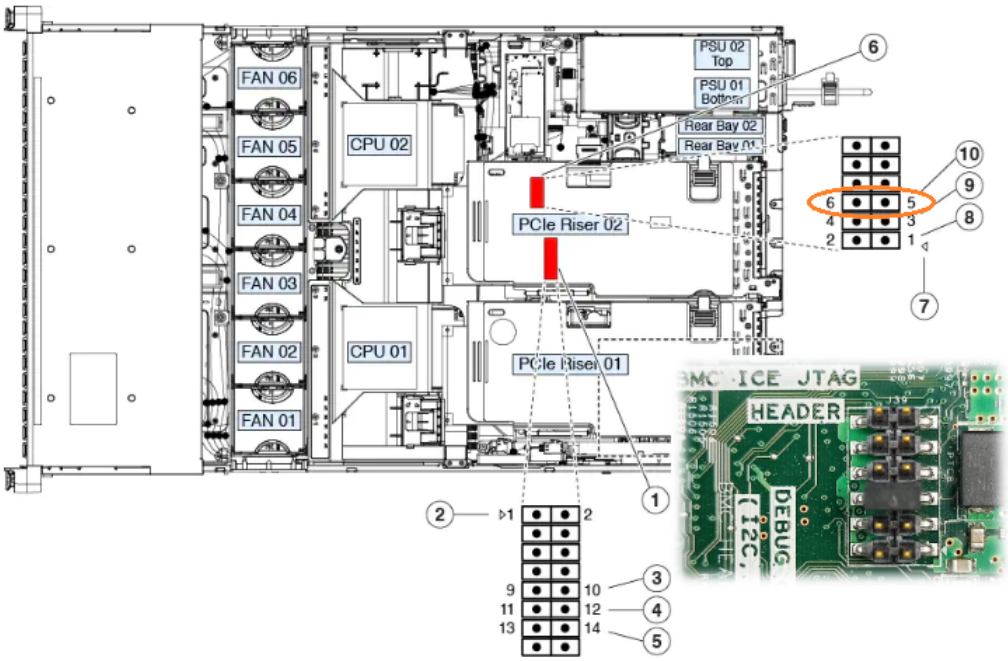
Хорошо … зайдём к этой проблеме с другой стороны.

Фактически сервер WSA S695, это ни что иное, как серверная платформа **Cisco Unified Computing System** (**UCS**) **C240 M5**. Эту маркировку мы можем обнаружить на верхней крышке этого сервера. От этого и будем отплясывать.

Внимательно изучаем руководство "Cisco UCS C240 M5 Server Installation and Service Guide" (https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/C240M5/install/C240M5.html). В главе "Maintaining the Server" находим раздел "Service Headers and Jumpers" (https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/C240M5/install/C240M5/C240M5_chapter_010.html#concept_vpx_jhp_jz) и обнаруживаем там описание размещения и контактных пинов группы **J39**.

Выключаем сервер, обесточиваем блоки питания, отключаем все провода и патч-корды. Выдвигаем сервер из серверной стойки и открываем верхнюю крышку. Ищем на материнской плате указанную группу пинов и ставим джампер на 5 и 6 пины, как показано на схеме:

Figure 48. Location of Service Header Blocks J38 and J39



1	Location of header block J38	6	Location of header block J39
2	J38 pin 1 arrow printed on motherboard	7	J39 pin 1 arrow printed on motherboard
3	Clear CMOS: J38 pins 9 - 10	8	Boot Cisco IMC from alternate image: J39 pins 1 - 2
4	Recover BIOS: J38 pins 11 - 12	9	Reset Cisco IMC password to default: J39 pins 3 - 4
5	Clear password: J38 pins 13 - 14	10	Reset Cisco IMC to defaults: J39 pins 5 - 6

Установка этой перемычки позволит сбросить все настройки контроллера CIMC в значения по умолчанию при следующем включении сервера.

Закрываем крышку сервера и возвращаем его в стойку.

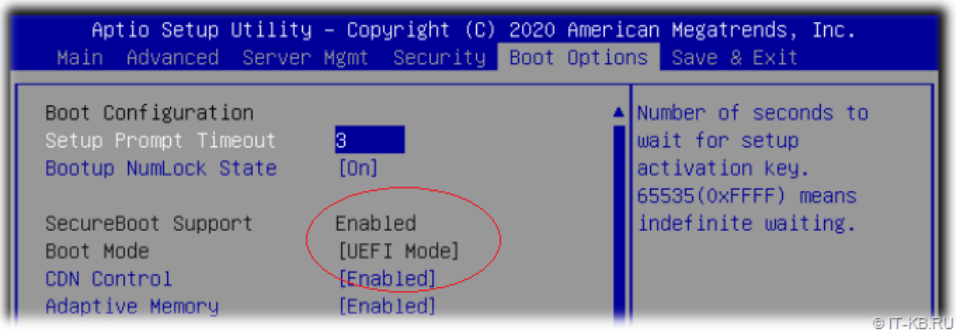
Теперь нам нужно обеспечить себе доступ к консоли сервера. На задней части сервера есть закрытый заглушкой **VGA** порт. С помощью правильной крестовой отвёртки аккуратно откручиваем заглушку (видимые нам наружные болты закручены не в корпус сервера, а прямо в фиксаторы разъёма D-Sub и поэтому могут прокручиваться). Подключаем монитор к открытому VGA порту. К расположенному рядом **USB**-порту подключаем клавиатуру.

Включаем сервер кнопкой питания на передней панели. Сервер после включения погудит вентиляторами какое-то время, не выдавая изображения на монитор, затем автоматически выключится. Ждём, ничего не предпринимаем. Примерно через минуту сервер включится повторно, снова погудит вентиляторами и затем снова выключится. Ещё минуту ждём следующего автоматического включения сервера. На этот раз он загрузится и на мониторе появится изображение загрузки системы.

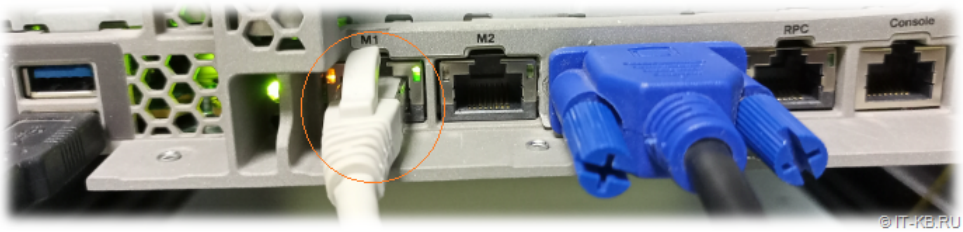
На этом этапе важно не позволить загрузиться операционной системе WSA. Для этого жмём кнопку "**F2**", чтобы нас перебросило в настройки BIOS.

В ходе загрузки убедимся в том, что адрес контроллера IMC указан, как "0.0.0.0". Это будет свидетельствовать о том, что IMC успешно сбросил свои настройки до состояния по умолчанию (особенно если мы ранее настраивали адрес IMC из AsyncOS с помощью команды "remotepower").

В настройках BIOS в разделе **Boot Options** мы увидим, что включен режим загрузки **UEFI**, активен **Secure Boot** и у нас нет возможности изменить эти настройки.

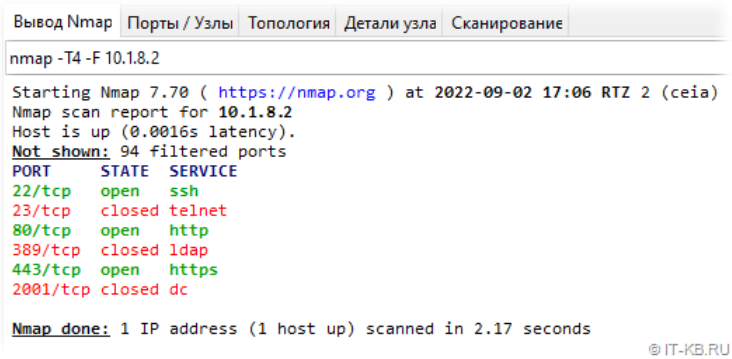


Теперь патч-кордом подключаем к локальной сети с **DHCP**-сервером порт сервера с маркировкой "**M1**":



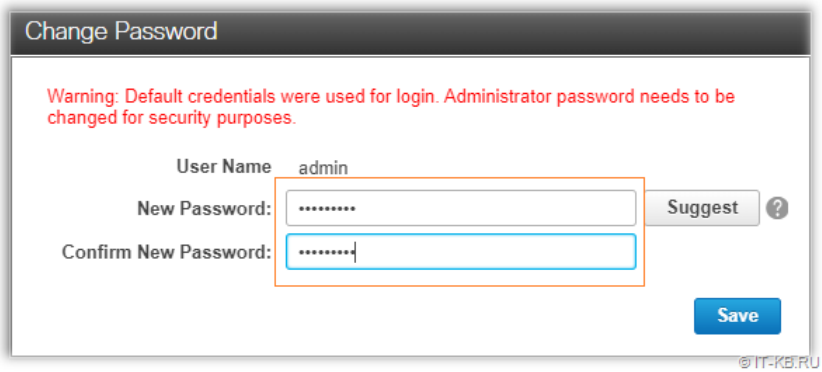
Ждём пару минут и проверяем DHCP-сервер на предмет появления нового арендованного адреса с именем типа "C240-<SN сервера>". Это наш сброшенный до дефолтных настроек контроллер IMC получил с DHCP новый IP-адрес.

Теперь можем проверить данный IP на предмет открытых портов с помощью быстрого сканирования утилитой **nmap**:

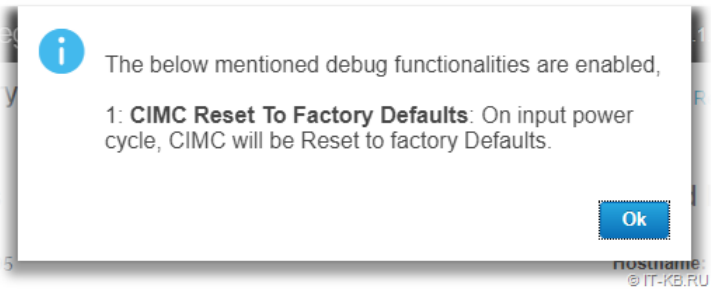


Как видим, нам стали доступны порты управления IMC по протоколам **SSH** и **HTTPS**. Пробуем подключится к IMC по протоколу SSH или через веб браузер. При первом входе используем учётные данные по умолчанию с логином "**admin**" и паролем "**password**".

После первого же входа мы получим всплывающее окно с требованием сменить пароль по умолчанию. Меняем пароль на новый и более сложный.



Перелогиниваемся и сразу видим всплывающее сообщение о том, что включён режим сброса настроек IMC, поэтому пока выполнять какую-либо конфигурацию не будем (следующая загрузка с включённым джампером снова затрёт все настройки IMC).



Теперь нам известен IP адрес IMC и заданы новые учётные данные для доступа к контроллеру.

Кнопкой питания или через интерфейс IMC выключаем сервер и затем обесточиваем его.

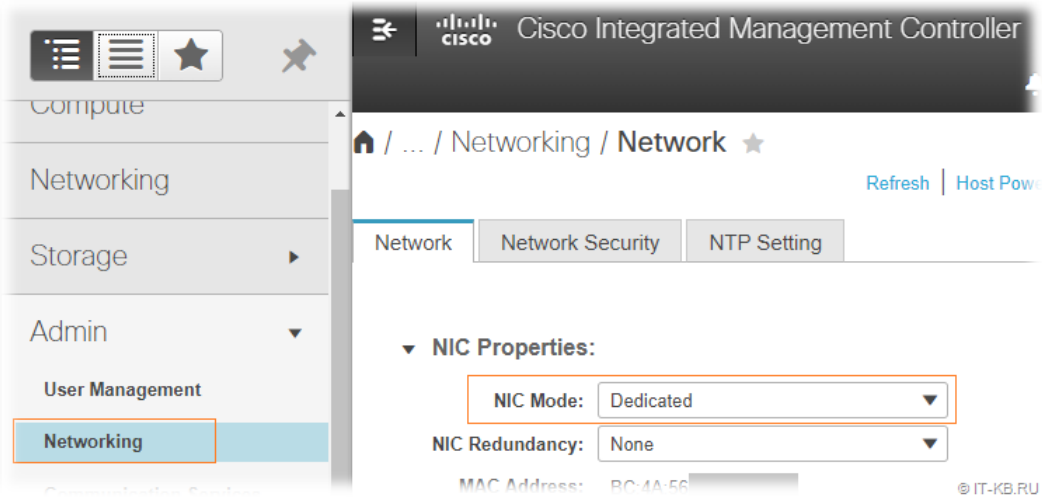
Убираем с материнской платы сервера ранее установленную перемычку с **J39**.

Затем включаем сервер кнопкой питания. При запуске сервер также несколько секунд пошумит вентиляторами, затем автоматически отключится и через минуту снова запустится. На этом этапе важно снова не дать серверу дойти до этапа загрузки ОС WSA (как и раньше, кнопкой "**F2**" входим в BIOS на этапе загрузки).

Смена режима работы интерфейса CIMC

В конфигурации по умолчанию в серверной платформе **UCS C240 M5** интерфейс контроллера **CIMC** работает в режиме "**Shared LOM Extende**", что не предполагает выделенного физического порта под функции управления. Это может быть актуально в случае нехватки физических портов под разные задачи и в этом случае интерфейс IMC может быть доступен через любой из активных портов, которые используются ОС под передачу данных. Такая конфигурация сама по себе накладывает дополнительные сложности и ограничения. В нашем случае физических портов более чем достаточно, поэтому предпочтительно выбрать вариант с использованием выделенного порта под функции управления IMC – режим "**Dedicated**".

Подключаемся на веб-интерфейс IMC, переходим в меню навигации в раздел "**Admin**" > "**Networking**" и на первой вкладке "**Network**" меняем режим работы интерфейса:



Внизу веб страницы жмём кнопку "**Save changes**" и принимаем сообщение о перезагрузке IMC в ходе применения изменений.

Переключаем патч-корд из порта "**M1**" в порт "**RPC**". Теперь это и будет выделенный порт для доступа к контроллеру CIMC.

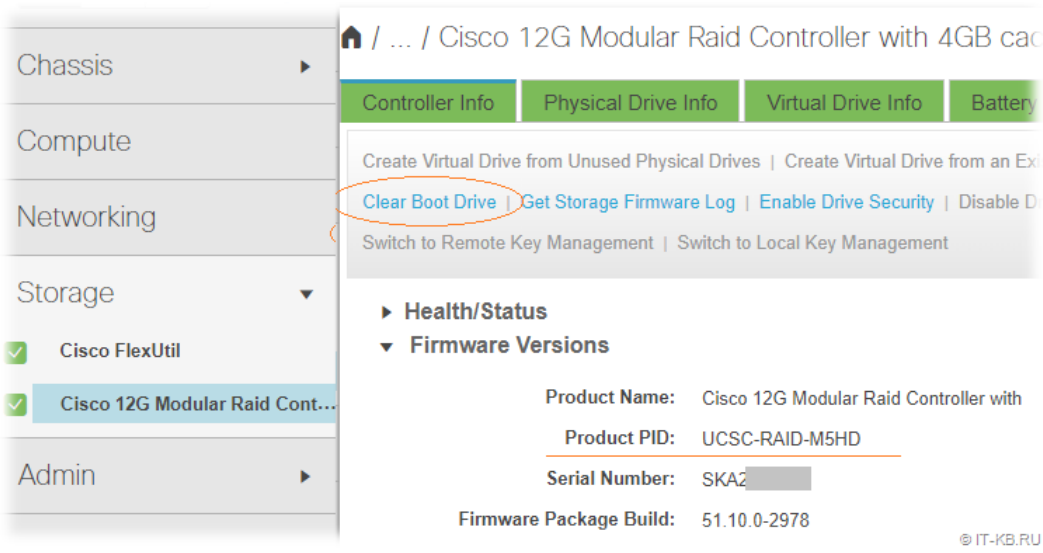
Может уйти несколько минут пока контроллер IMC перестроит сетевой интерфейс и он станет доступен. При переключении на практике на одном из серверов было замечено, что после изменения настроек и переключения порта, интерфейс так и не становился доступен на протяжении длительного времени. В этой ситуации помогло полное отключение/включение (с обесточиванием) всей серверной платформы.

Удаление RAID-диска с AsyncOS

Итак, мы получили полноценное управление контроллером CIMC и провели его некоторую первичную настройку. Но вся эта работа может сойти на нет, если мы снова позволим загрузиться AsyncOS, так как при первой же загрузке этой ОС автоматически отключатся протоколы полного доступа к IMC, такие как SSH и HTTPS.

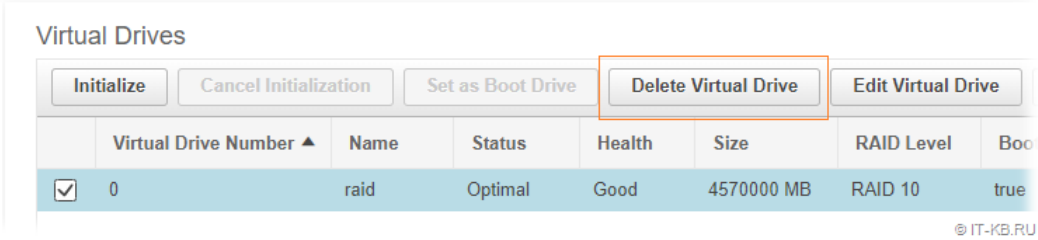
Учитывая то, что, как таковая, AsyncOS без действующей лицензионной подписки нам бесполезна, мы просто помножим на ноль эту дивную ОС, разобрав RAID-диск, на котором она размещена.

Для управления RAID в веб-консоли IMC переходим в раздел "**Storage**", выбираем "**Cisco 12G Modular Raid Controller**" и на первой вкладке "**Controller Info**" выбираем пункт отключения флага загрузочного диска "**Clear Boot Drive**" (без этого IMC не позволит нам удалить RAID-диск):

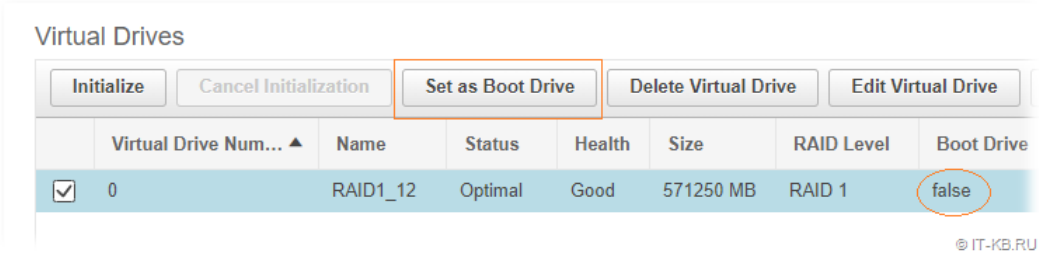


Заодно обратим внимание на **Product ID** нашего RAID-контроллера. Эта информация может пригодится нам в дальнейшем при работе с драйверами под разные ОС.

Перейдём на вкладку "**Virtual Drive Info**" и удалим виртуальный RAID диск **vd-0**.



Теперь здесь же мы можем создать новые RAID-диски таким образом, как этого требуют наши задачи. При создании диска под новую ОС не забываем включать признак того, что этот диск загрузочный.



Кстати, практические опыты показали, что RAID-контроллер не имеет ярко выраженных признаков "вендор-лока" по накопителям. По крайней мере я без особых проблем в 1-2 слоты дисковой корзины серверов установил сторонние SAS SSD накопители от HGST и WD, собрал из них RAID-1 и в дальнейшем установил на него нужную мне ОС.

Обновление микрокода Cisco UCS. Общие понятия

Для обновления микрокода BIOS, IMC и других аппаратных компонент сервера Cisco UCS нам потребуется скачать загрузочный образ с инструментом "**Cisco UCS Host Upgrade Utility**" (**HUU**). Архив с разными версиями этого образа можно найти по ссылке: "UCS Server Firmware" (https://software.cisco.com/download/home/286318800/type/283850974/release/4.2(2a)).

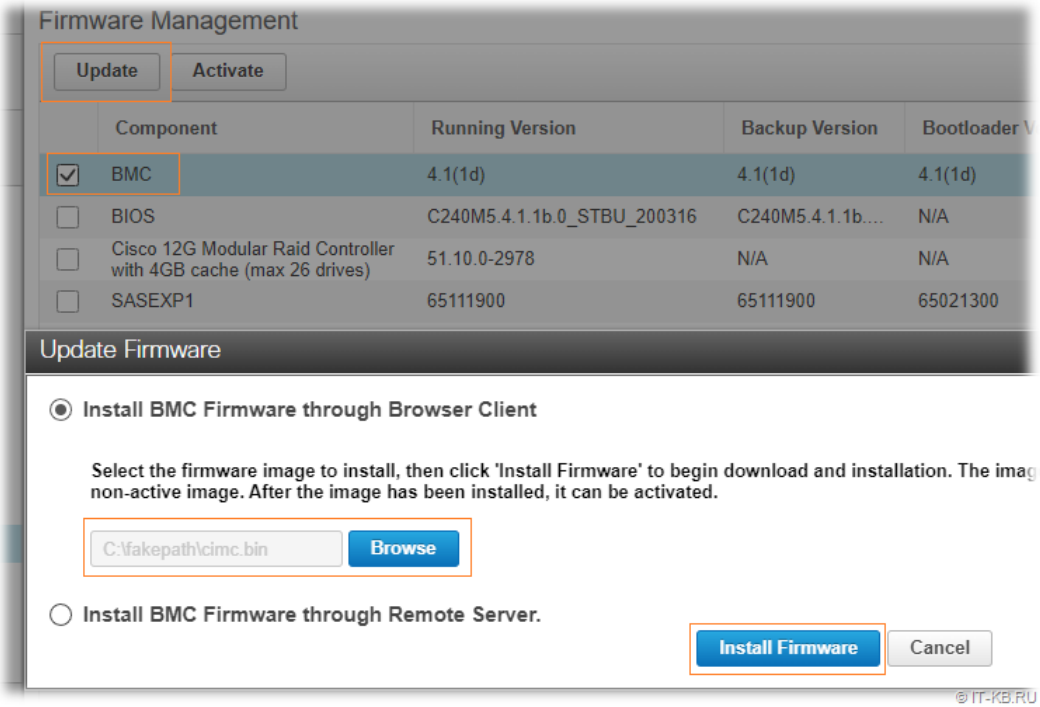
Здесь может встать вопрос в том, как ориентироваться во множестве версий HUU. Логичным выглядит выбор HUU самой новой версии, но это не совсем неприменимо в нашей ситуации. Нам потребуется несколько разных версий HUU. Более новые версии мы можем использовать для обновления микрокода контроллера IMC и других серверных компонент типа RAID-контроллера, сетевых адаптеров, накопителей и т.д.. Более старые версии HUU нам понадобятся, чтобы решить проблему с неуправляемым Secure Boot.

Сейчас нас в первую очередь интересуют прошивки IMC и BIOS, в версионности которых есть некоторые нюансы. Как я понял, версия BIOS не должна быть новей версии IMC. Например, если взять базовую поставку наших серверов WSA S695, то мы увидим версию BIOS 4.1.1b при версии IMC 4.1(1d). Полагаю, что если выбирать версию IMC, то мы можем взять её микрокод из самой актуальной версии HUU. Контроллер IMC представляет разные программные интерфейсы удалённого управления серверной платформой и поэтому, с точки зрения безопасности, более логичным будет использование самой актуальной версии микрокода IMC с поддержкой последних версий протоколов и механизмов аутентификации. При этом следует понимать, что часть функционала в обновлённой версии IMC может не работать со старыми версиями BIOS.

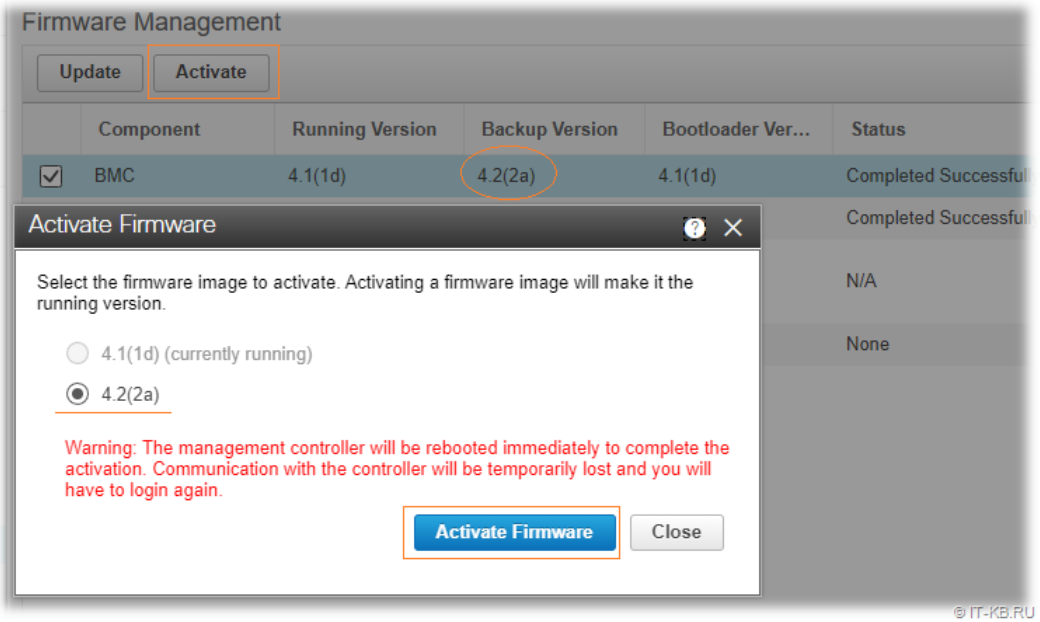
Обновление микрокода CIMC

Итак, мы условились, что будем использовать самую актуальную версию прошивки IMC и, соответственно, нам нужен образ HUU версии 4.2(2a) 08-Jul-2022 (https://software.cisco.com/download/home/286318800/type/283850974/release/4.2(2a)) (файл ucs-c240m5-huu-4.2.2a.iso). Как отмечалось ранее, образ HUU является загрузочным и представляет удобный интерфейс для обновления микрокода всех аппаратных компонент типичного сервера UCS. Но в нашей ситуации возможности сервера искусственно ограничены жесткими настройками Secure Boot и поэтому мы банально не сможем загрузить образ HUU, впрочем, как и любой другой загрузочный образ. Поэтому мы решим вопрос обновления микрокода IMC с помощью самого IMC.

Переходим в веб-интерфейсе IMC в раздел "**Admin**" > "**Firmware Management**", выбираем "**BMC**" и нажимаем "**Update**". В открывшейся модальной форме выбираем файл **cimc.bin** из предварительно распакованного образа HUU (файл можно найти в подкаталоге \cisco_firmware\cimc образа ucs-c240m5-huu-4.2.2a.iso):

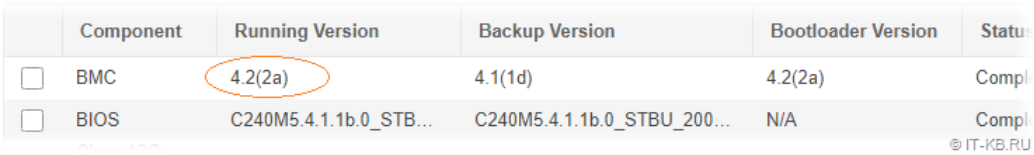


Дожидаемся окончания верификации, загрузки и установки новой версии прошивки. Загруженная версия будет помечена как "**Backup Version**" и для её активации жмём "**Activate**":



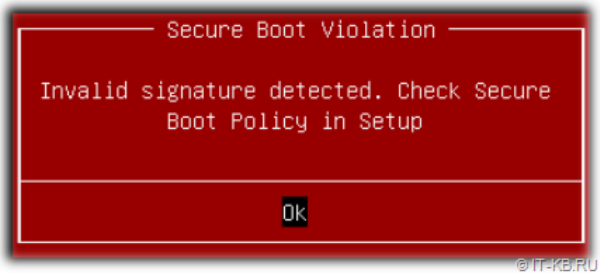
В ходе активации прошивка "**Backup Version**" перейдёт в стадию "**Running Version**", а версия, которая была активной ранее, напротив, перейдёт в состояние "**Backup Version**".

Контроллер IMC в ходе переключения версии микрокода перезагружается и будет недоступен какое-то время. Переподключаемся к веб-консоли IMC и проверяем текущую активную версию микрокода.



Замена (даунгрейд) микрокода BIOS

Как отмечалось ранее, текущая версия BIOS не позволяет нам при необходимости изменять тип загрузки с UEFI на Legacy и имеет жёстко заданный UEFI Mode с включённым Secure Boot. Это приводит к тому, что любые попытки загрузить сервер со стороннего накопителя для установки какой-либо альтернативной ОС будут приводить нас к ошибке **Secure Boot Violation** "Invalid signature detected..."



Опытный перебор множества доступных на текущий момент на сайте Cisco прошивок BIOS показал, что такие настройки в "**Boot Options**" были не во всех версиях.

При даунгрейде версии **BIOS** до ветки **4.0.1** обнаруживается, что переключатель режима загрузки "Boot Mode" становится доступным для изменения, но Secure Boot при этом по прежнему включён и недоступен для изменения. При этом выбор режима Legacy является на самом деле безрезультативным, так как после перезагрузки микрокод снова автоматически возвращается в режим UEFI с Secure Boot.

И лишь при даунгрейде **BIOS** до ветки **3.1.3** выключается форсированное применение режима Secure Boot и действительно появляется работающий выбор между режимами загрузки Legacy Mode и UEFI Mode. Но при скачивании пакетов утилиты HUU этой ветки с сайта Cisco нужно учесть один нюанс. Дело в том, что последней предлагаемой версией HUU из ветки 3.1.3 является версия 3.1(3k) 21-Sep-2020 (https://software.cisco.com/download/home/286318800/type/283850974/release/3.1(3k)), то есть образ с именем ucs-c240m5-huu-3.1.3k.iso. На самом деле в этот образ включена прошивка BIOS уже из ветки 4.0.4 и эта прошивка имеет форсированно включенный Secure Boot. Испытание прошивок из HUU ветки 3.1.3 показали, что максимально актуальной версией с выключенным Secure Boot и возможностью выбора между режимами UEFI/Legacy является версия 3.1(3j) 17-Sep-2019 (https://software.cisco.com/download/home/286318800/type/283850974/release/3.1(3j)), то есть образ с именем ucs-c240m5-huu-3.1.3j.iso. Внутри этого образа можно найти микрокод BIOS версии **3.1.3h.0** Build 08/10/2019. Вот он то нам и нужен.

Прошиваем BIOS по аналогии с тем, как было показано ранее, через веб-интерфейс IMC в разделе "**Admin**" > "**Firmware Management**". На этот раз используем файл **bios.cap** из распакованного образа HUU из подкаталога \cisco_firmware\bios.

Однако выполнить даунгрейд предустановленной версии BIOS 4.1.1b сразу до версии 3.1.3h у нас не получится. При попытке загрузки файла микрокода мы получим сообщение "Update aborted. BIOS version incompatible with installed processor". Чтобы решить эту проблему, нам потребуется провести даунгрейд BIOS в несколько этапов. У меня получилось добиться желаемого результата следующей последовательностью действий:

1) Даунгрейд **BIOS 4.1.1b** -> **4.0.4s**

2) Даунгрейд **BIOS 4.0.4s** -> **4.0.2f**

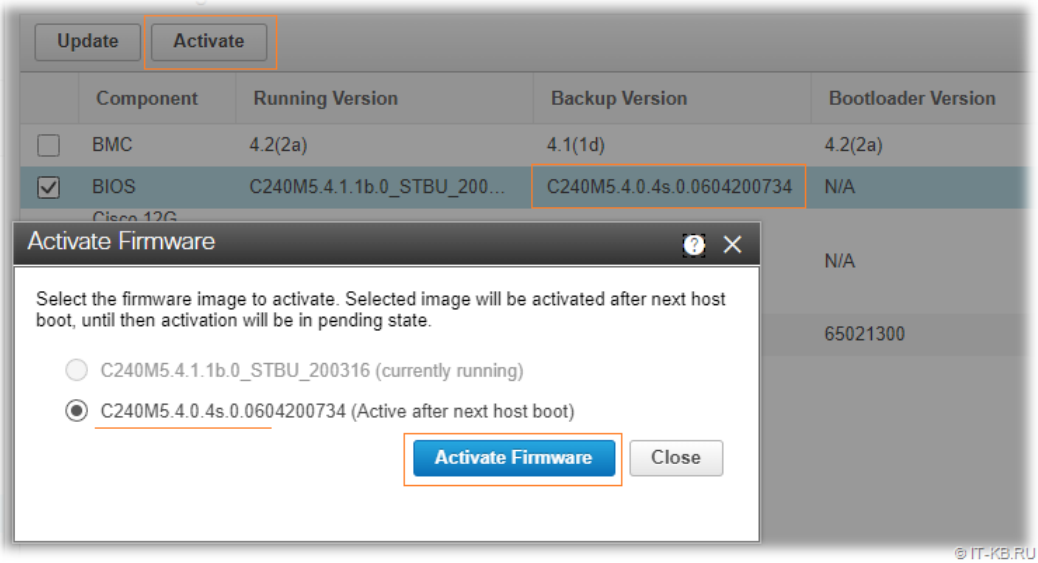
3) Понижаем версию **IMC** с **4.2.2a** до **4.0.2n**

4) Даунгрейд **BIOS 4.0.2f** -> **3.1.3h**

5) Опционально. Повышаем версию **IMC** с **4.0.2n** на **4.2.2a**.

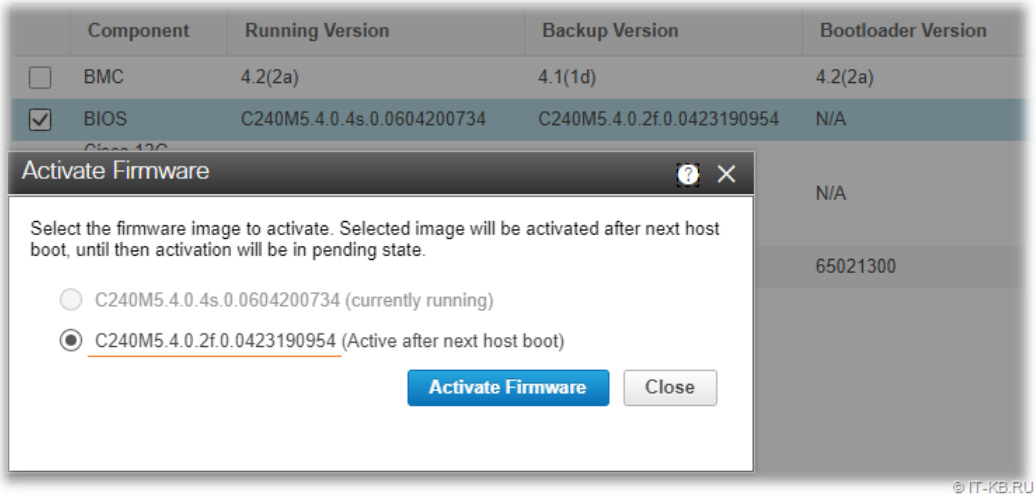
Пробежимся по этим пунктам.

На первом шаге мы понизим имеющуюся у нас **BIOS** с **4.1.1b** до **4.0.4s**, используя файл bios.cap из образа HUU 3.1(3k) 21-Sep-2020 (https://software.cisco.com/download/home/286318800/type/283850974/release/3.1(3k)) (ucs-c240m5-huu-3.1.3k.iso). После успешной загрузки микрокода выполняем его активацию.



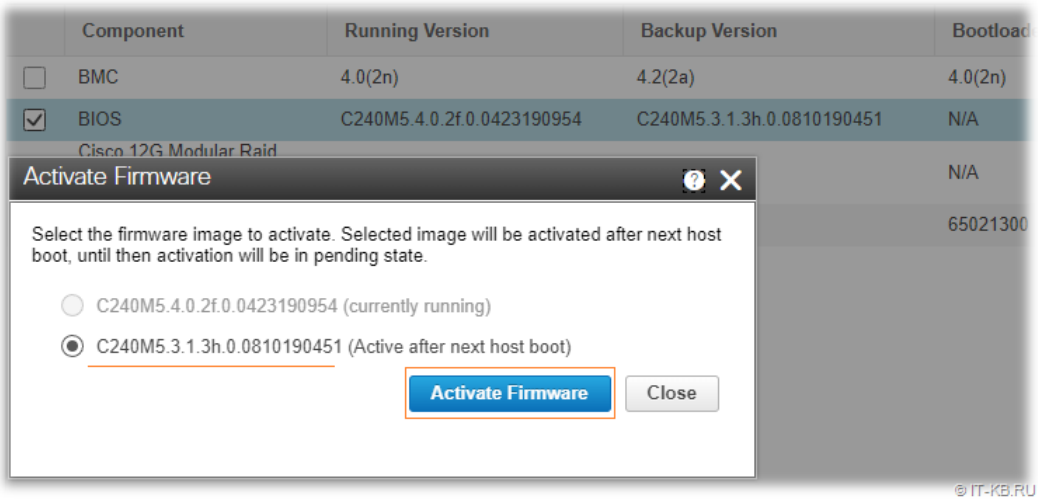
При активации прошивка перейдёт в стадию "**Activation Pending**". Чтобы окончательно активировать новую версию потребуется перезагрузка сервера.

На втором шаге, после того как сервер загружен с **BIOS 4.0.4s**, снова с помощью веб-интерфейса IMC выполняем даунгрейд до следующей по цепочке версии **4.0.2f**, используя файл bios.cap из HUU 4.0.2n 16-Aug-2020 (https://software.cisco.com/download/home/286318800/type/283850974/release/4.0(2n)) (образ с именем ucs-c240m5-huu-4.0.2n.iso). Активируем версию и выполняем перезагрузку сервера.

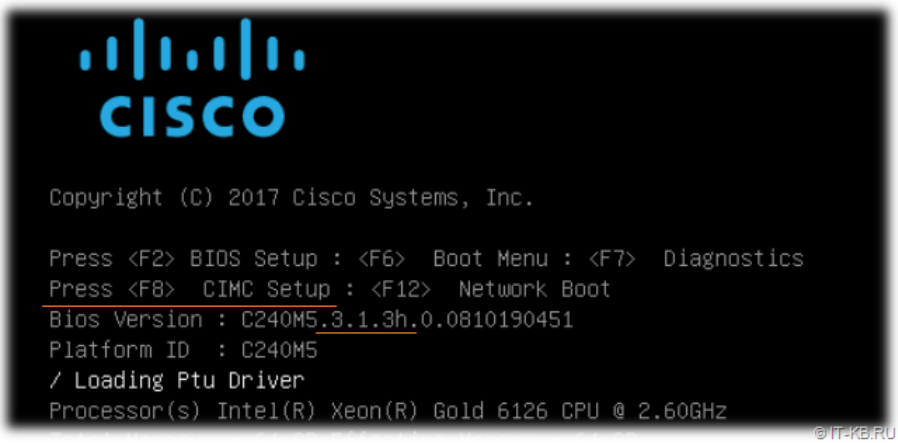


Третий шаг является вынужденной мерой так, как обновлённая версия IMC 4.2.2a при попытке выполнить даунгрейд BIOS в четвёртом шаге будет возвращать нам уже знакомую ошибку "Update aborted. BIOS version incompatible with installed processor". Поэтому, чтобы избежать этой проблемы, выполняем понижение версии **CIMC** с **4.2.2a** до **4.0.2n**, используя файл cimc.bin из HUU 4.0.2n 16-Aug-2020 (https://software.cisco.com/download/home/286318800/type/283850974/release/4.0(2n)) (образ с именем ucs-c240m5-huu-4.0.2n.iso). Активируем пониженную версию IMC и дожидаемся перезагрузки контроллера после активации.

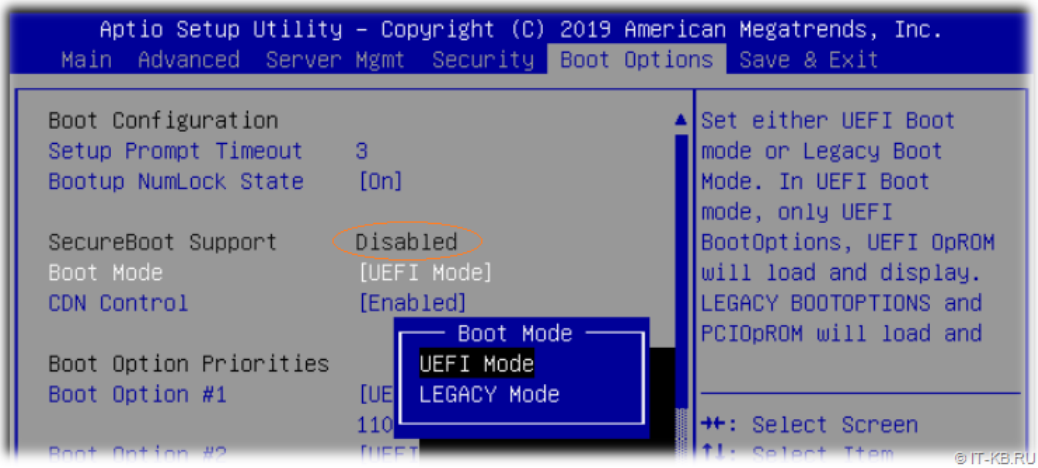
На четвёртом шаге мы выполняем окончательный даунгрейд **BIOS** с версии **4.0.2f** до нужной нам версии **3.1.3h**, используя файл bios.cap из HUU 3.1(3j) 17-Sep-2019 (https://software.cisco.com/download/home/286318800/type/283850974/release/3.1(3j)) (образ с именем ucs-c240m5-huu-3.1.3j.iso). Активируем версию и выполняем перезагрузку сервера.



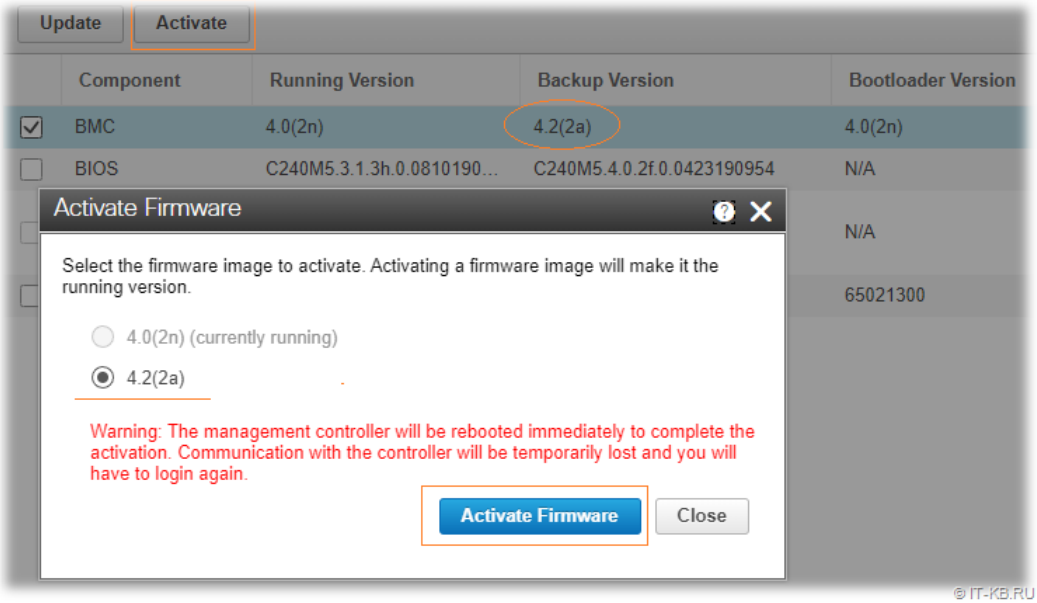
В ходе загрузки мы увидим, что теперь нам стала доступна функция входа в TUI-инструмент управления IMC по кнопке "**F8**".



Зайдём в настройки BIOS и обнаружим, что теперь **Secure Boot** выключен и у нас есть возможность менять режим загрузки **UEFI**/**Legacy**.



По поводу пятого шага, напомню, что в ходе даунгрейда BIOS мы немного снизили версию **IMC**. С точки зрения IMC текущая (пониженная) версия IMC 4.0.2n "родней" к выбранной нами старой версии BIOS. Но, опять же, с точки зрения безопасности, можно подняться на новую версию IMC. Тем более, поднять версию мы можем путём простого переключения с версии **4.0.2n** на версию **4.2.2a**, так как обе эти версии есть у нас в IMC. Для этого достаточно просто выполнить активацию "**Backup Version**" для BMC:

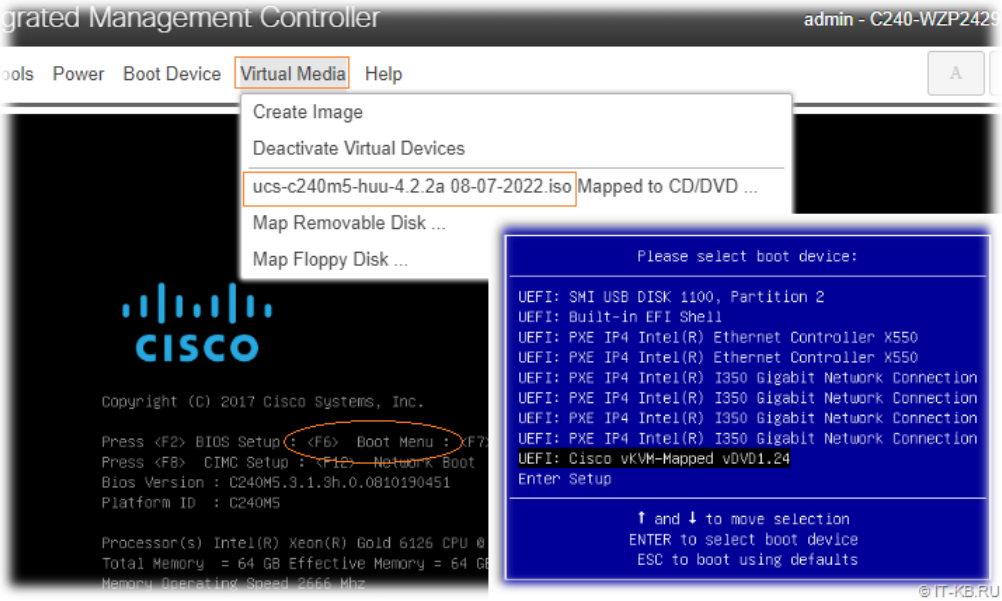


На этом танцы с микрокодом IMC и BIOS можно считать завершёнными.

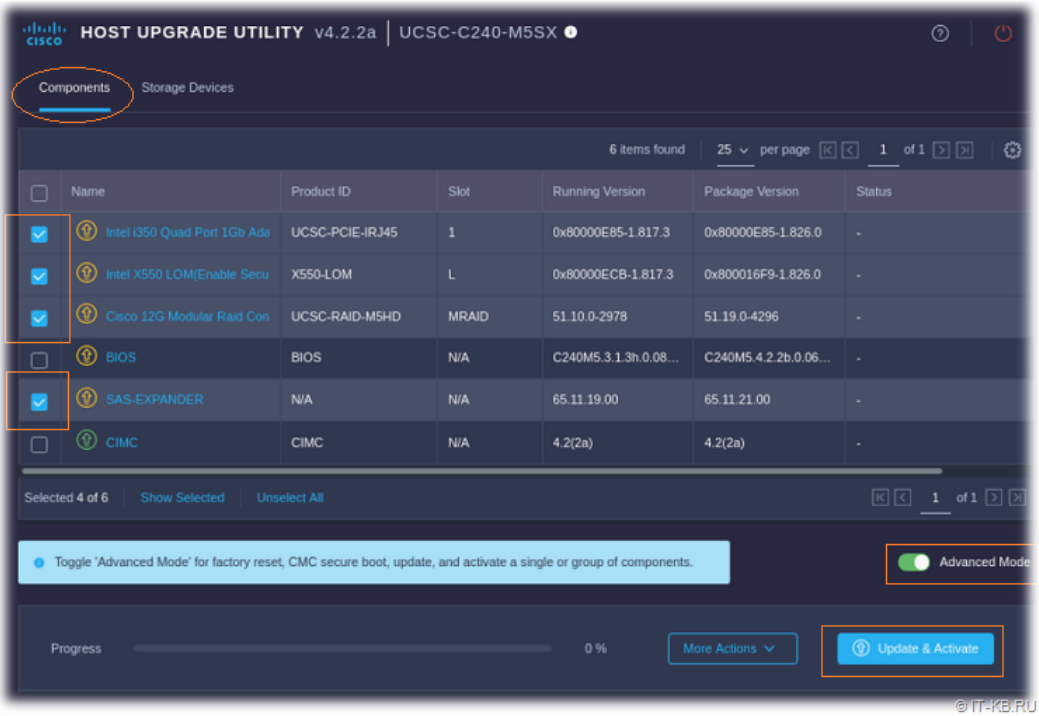
Опциональное обновление микрокода устройств

Так как теперь у нас появилась возможность загружать сервер с любых загрузочных накопителей, воспользуемся загрузочным образом последней версии **HUU** для актуализации микрокода сетевых адаптеров, RAID контроллера, дисковых накопителей и других устройств.

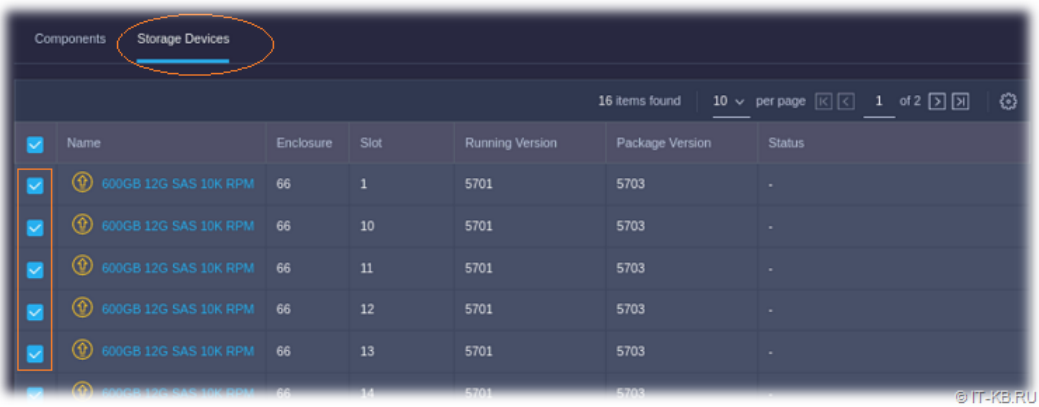
Смонтируем в виртуальный привод веб-консоли **CIMC vKVM** образ актуальной версии HUU 4.2(2a), перезагрузим сервер и в ходе загрузки воспользуемся загрузочным меню по кнопке "**F6**":



Дожидаемся загрузки графической оболочки HUU и включаем в ней расширенный режим "**Advanced Mode**", чтобы получить возможность выбора компонент, которые требуется обновить. При этом помним, что на вкладке "**Components**" включать обновление BIOS в нашей ситуации не следует, так как новая версия микрокода опять форсировано задействует механизм Secure Boot и мы вернёмся к тому, с чего начали.



На вкладке "**Storage Devices**" мы можем разрешить обновление микрокода всех совместимых дисковых накопителей. При этом замечено, что даже если выбраны все накопители, прошивка может пройти не полностью, а только для тех дисков, которые выбраны на первом экране. Поэтому лучше сразу включить режим отображения большего количества дисков на одной вкладке.



По завершении процедуры обновления и перезагрузки сервера не лишним будет повторно запустить HUU, чтобы убедиться в том, что все интересующие нас обновления применены успешно.

Установка желаемой операционной системы

Всё что нам теперь остаётся, это беспрепятственно провести установку желаемой операционной системы на наш бывший сервер Cisco WSA. В ходе установки нам может потребоваться дополнительная подгрузка драйверов, отсутствующий в базовом составе устанавливаемой ОС. Так, например, при установке на сервер ОС **Windows Server 2016** мне потребовалось инсталлятору ОС подмонтировать через **vKVM** образ с драйвером от **RAID** контроллера, чтобы инсталлятор смог обнаружить RAID диск (подкаталог \Storage\LSI\UCSC-RAID-M5\W2K16\ с образа ucs-cxxx-drivers-windows.4.2.2c.iso (https://software.cisco.com/download/home/286318800/type/283853158/release/4.2(2c))). После установки ОС этот же образ можно использовать для развёртывания в системе всех недостающих драйверов.

По итогу у нас не только появилась возможность развёртывания любой операционной системы "по вкусу" но и возможность полноценно управлять этой серверной платформой через встроенный контроллер Cisco IMC. И, как следствие, у нас появилась возможность дальнейшей настройки мониторинга состояния аппаратной части подобных серверов через протоколы SNMP/SSH/XML API.

14 Оценок

Опубликовано в : Cisco (https://blog.it-kb.ru/category/cisco/) , Hardware (https://blog.it-kb.ru/category/hardware/)

Метки : AsyncOS (https://blog.it-kb.ru/tag/asyncos/) , BIOS (https://blog.it-kb.ru/tag/bios/) , BMC (https://blog.it-kb.ru/tag/bmc/) , Boot (https://blog.it-kb.ru/tag/boot/) , CIMC (https://blog.it-kb.ru/tag/cimc/) , Cisco (https://blog.it-kb.ru/tag/cisco/) , Cisco IMC (https://blog.it-kb.ru/tag/cisco-imc/) , Cisco UCS (https://blog.it-kb.ru/tag/cisco-ucs/) , Cisco WSA (https://blog.it-kb.ru/tag/cisco-wsa/) , Downgrade (https://blog.it-kb.ru/tag/downgrade/) , Drivers (https://blog.it-kb.ru/tag/drivers/) , firmware (https://blog.it-kb.ru/tag/firmware/) , Hardware (https://blog.it-kb.ru/tag/hardware/) , HUU (https://blog.it-kb.ru/tag/huu/) , IMC (https://blog.it-kb.ru/tag/imc/) , IPMI (https://blog.it-kb.ru/tag/ipmi/) , KVM (https://blog.it-kb.ru/tag/kvm/) , RAID (https://blog.it-kb.ru/tag/raid/) , RPC (https://blog.it-kb.ru/tag/rpc/) , Secure Boot (https://blog.it-kb.ru/tag/secure-boot/) , UCS (https://blog.it-kb.ru/tag/ucs/) , UCS C240 M5 (https://blog.it-kb.ru/tag/ucs-c240-m5/) , UEFI (https://blog.it-kb.ru/tag/uefi/) , Upgrade (https://blog.it-kb.ru/tag/upgrade/) , vKVM (https://blog.it-kb.ru/tag/vkvm/) , Windows Server 2016 (https://blog.it-kb.ru/tag/windows-server-2016/) , WSA S695 (https://blog.it-kb.ru/tag/wsa-s695/)

Всего комментариев: 3

Комментировать

Павел / 08.09.2022 10:34

Это шедевр!

Ответить

Андрей Вахитов / 14.09.2022 20:00

Снимаю шляпу

Ответить

Алексей Максимов (https://blog.it-kb.ru/author/blogroot/) / Автор записи 14.09.2022 21:14

Это ещё цветочки. Ягодки начались, когда я взялся за платформу M4... приключения с M5 мне показались "легкой прогулкой". Следующая статья будет про это.

Ответить

Добавить комментарий

Введите свой комментарий...

Социальные ссылки

Email: Blog@IT-KB.RU (mailto:Blog@IT-KB.RU)

(https://blog.it-kb.ru/feed/)

(https://twitter.com/Blog_IT_KB)

(https://www.facebook.com/blog.it.kb)

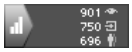
(https://blog-it-kb.tumblr.com/)

(https://vk.com/blogitkb)

Защита SSL

(https://www.gogetssl.com)

Статистика



(https://metrika.yandex.ru/stat/?id=37925125&from=informer)